
Analysis

for

Metamodel tool

Version 2.0 approved

Vrchlpet

MMN group

6.12.2010

1 Table of Contents

1	Table of Contents	2
1.	Uvod.....	3
2.	Specifikace	3
3.	Use Case diagram	5
4.	Data flow diagram	6
5.	Class diagram.....	7
6.	Diagram aktivit	9

1. Uvod

MMN je nástroj pro vytváření metamodelů diagramů. Hlavním cílem projektu je, poskytnout uživateli nástroj, ve kterém si vytvoří sémantiku svého diagramu, kterou poté bude moci použít v notační aplikaci (není součástí projektu), ve které bude podle pravidel danými metamodelem vytvářet různorodé diagramy. Předností MMN projektu je uživatelská přívětivost, která mimo jiné zahrnuje co možná nejsnazší ovládání pomocí grafického rozhraní.

MMN bude sloužit jako výplň mezery mezi existujícími notacemi a to ať už UML diagramy, různými diagramy pro modelování business procesů, aktivit nebo scénářů. Uživateli bude nabídnuta možnost vytvořit si vlastní notaci, která přesně pokryje jeho požadavky na modelování určité problematiky.

2. Specifikace

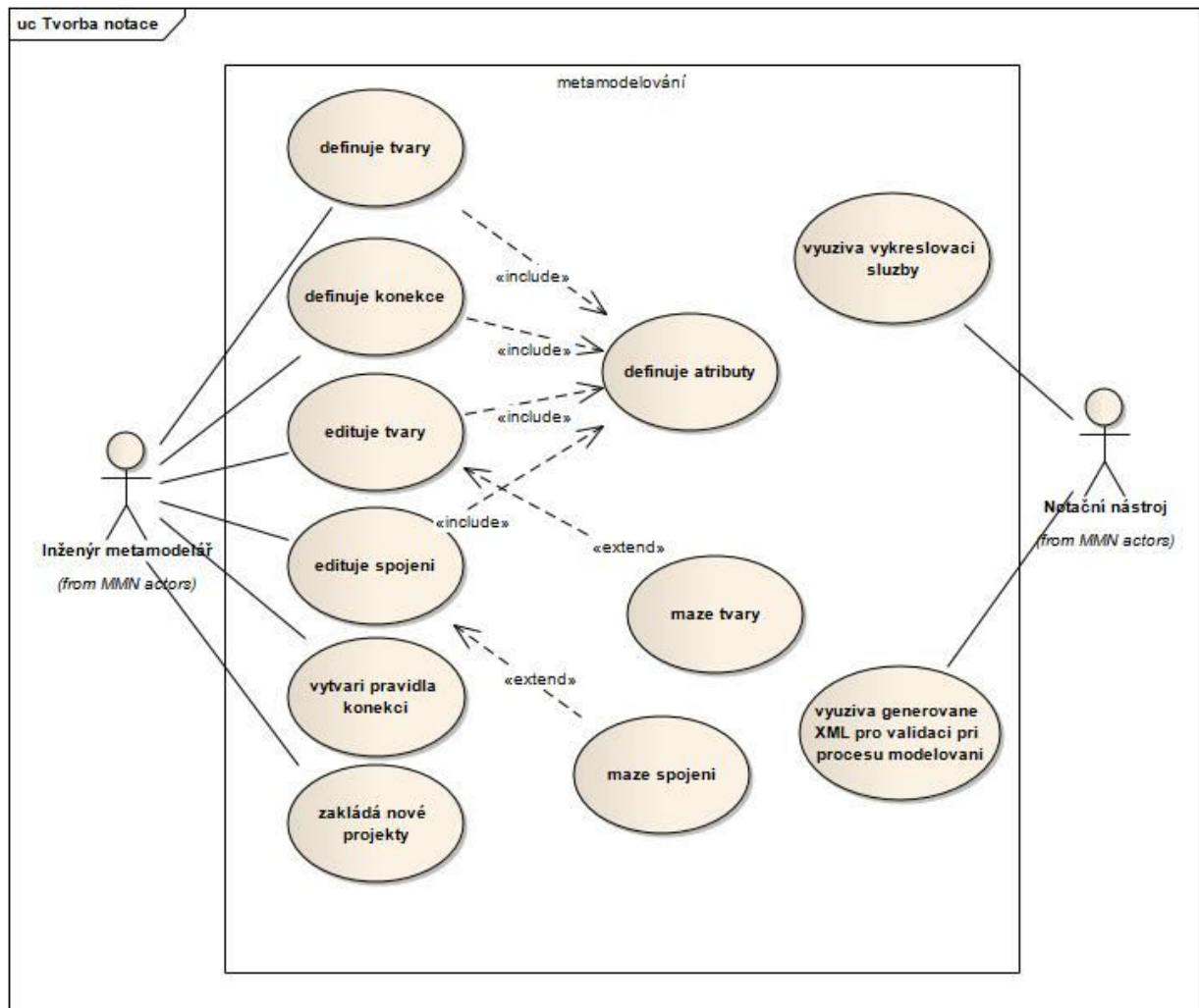
Níže uvedená tabulka popisuje specifikace projektu, které byly sbírány, analyzovány a specifikovány téměř po celou dobu vývoje aplikace.

SRS id	SRS description	SRS priority
FUNCTIONAL REQUIREMENTS		
0.0	MMN is a tool for creating metamodels of notations and will provide services, which will help other applications to paint objects(shapes, connections etc...). Any metamodel can be saved and load for further modeling.	must have
1.0	MMN will allow to define connections	must have
1.1	MMN will allow to define connection's attributes	must have
1.2	Obligatory attributes of connection is connection's unique name	must have
1.3	Optional attributes are source and target type of arrow	must have
1.4	Other optional attributes are connection's labels	nice to have
1.5	MMN will group connections in categories	must have
1.6	MMN will provide default category	should have
1.7	Default category will contain association and generalization as a default connections of each new created metamodel	should have
1.8	MMN will also provide a custom category, which will serve as a space for new, by user customized, connections	must have
1.9	Individual connections can be draw on workspece by dragging them from palette and dropping them onto workspace. This action will draw an table with a view of the connection. The table will provide functionality for creating rules. To create a rule, user will choose source and target shape and will define their multiplicity.	must have
1.10	Shapes can participate in rules only if they already exist in metamodel	must have
1.11	Valid multiplicity values are: * for random multiplicity(0 to infinite) or any number greater or equal to zero, which will represent a concrete multiplicity rule.	must have
1.12	New connections can be created by defining them in dialog window	must have
1.13	In dialog, there will be options for setting name and attributes of new connections	must have

1.14	Connections can be deleted	must have
1.15	Connections can be edited	must have
2.0	MMN will allow to define shapes	must have
2.1	MMN will allow to define shape's attributes	must have
2.2	Obligatory attribute of shape is its unique name	must have
2.3	Optional attributes of shape are labels, images and tooltips	must have
2.4	Other optional attributes of shapes are editable text fields	nice to have
2.5	MMN will group shapes in categories	must have
2.6	MMN will provide default category	should have
2.7	The default category will contain basic shapes such as circle or rectangle	should have
2.8	MMN will also provide a custom category, which will serve as a space for new, by user customized, shapes	must have
2.9	Individual shapes can be draw on workspace by dragging them from palette and dropping them onto workspace.	must have
2.10	New shape can be created by defining them in dialog window	must have
2.11	In dialog there will be options for settings name and attributes of new shapes	must have
2.12	Shapes can be deleted	must have
2.13	Shapes can be edited	must have
3.0	MMN will implement a new file type for metamodels which will be usable in netbeans platforme	must have
4.0	MMN will support projects creations	must have
4.1	New metamodel cen be created only under specific project	must have
5.0	Projects can be load or saved as a XML files	must have
6.0	MMN will provide projects browser	must have
6.1	By selecting projects and its metamodel, MMN will open an editor, where can be metamodel designed	must have
6.2	For each metamodel there will be a multiview consist of 2 views. First view is for defining rules, second view is for creating, editing and rendering shapes	must have
7.0	MMN will provide an xml generator, which will generate a xml file with notation rules. This file will be used for notations plugins as a source of notation rules. Notation plugins will use this rules for validating specific notation and for asking MMN for painting objects for them.	must have
8.0	MMN will provide an interface, which will serve as a object factory for creation of shapes and connections according to the xml file generated by MMN. Notation plugins will use this interface as a painting tool	must have
NON-FUNCTIONAL REQUIREMETNS		
1.0	MMN will be programmed in Java	must have
2.0	MMN will be a plugin for NetBeans platform	must have
3.0	MMN will use Visual library API(NetBeans platforme module) for graphical representation of metamodels	must have

3. Use Case diagram

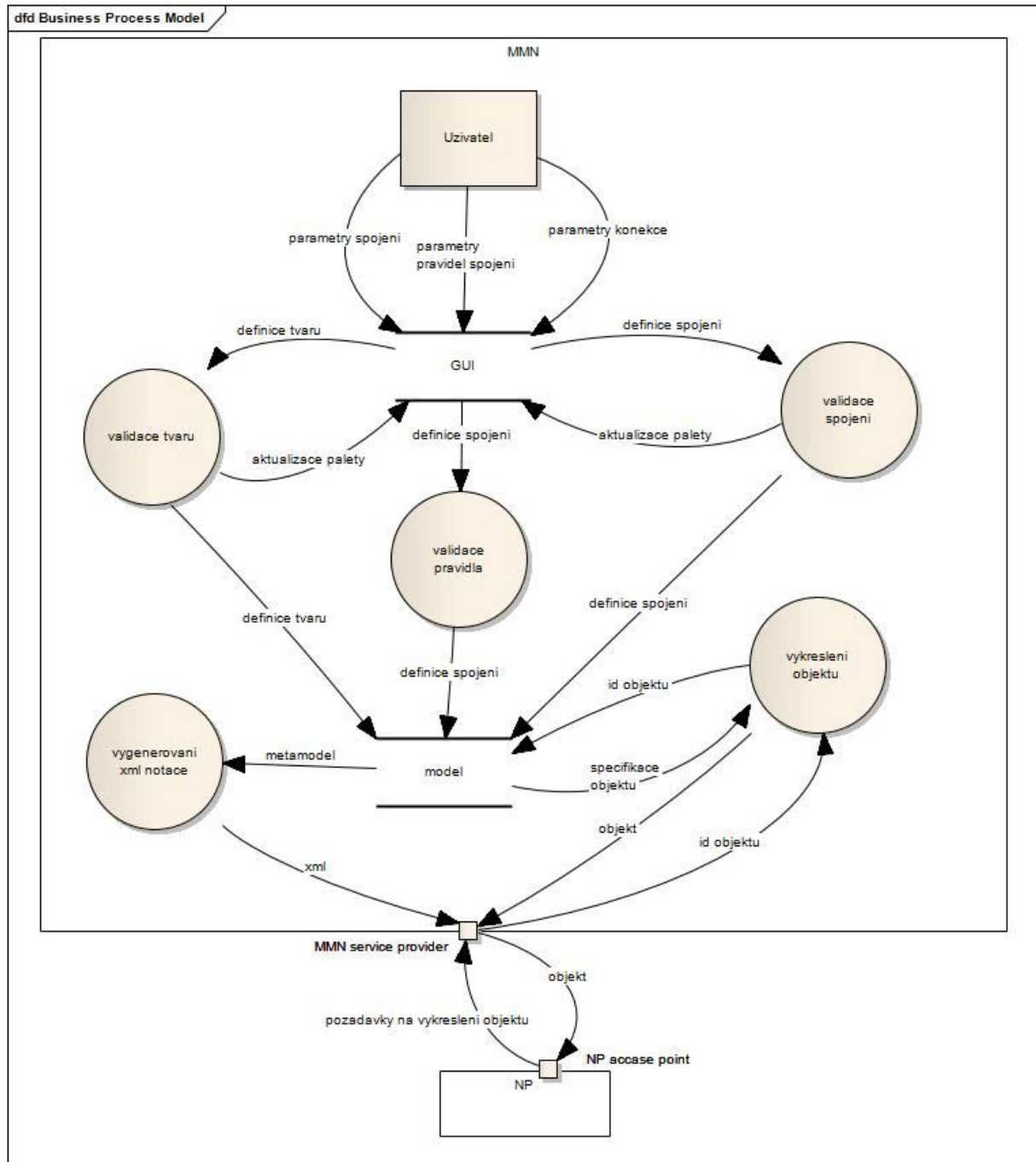
Use case diagram znázorňuje role aktorů vůči aplikaci. MMN definuje 2 aktéry. Konkrétně aktéra Inženýr metamodelář, který je vlastně uživatel MMN, který používá pro definování metamodelu notace. Druhý aktor je notační nástroj. Notační nástroj (notační plugin NP) je plugin pro NetBeans platformu, který využívá XML soubor generovaný MMN, ve kterém jsou obsaženy data o metamodelu notace, včetně pravidel konekcí, tj. jaký tvar může být spojený s jakým tvarem pomocí jaké konekce. Tento XML soubor slouží pro validaci diagramu tvořeném pomocí NP a pro posílání požadavků na rozhraní MMN pluginu na vykreslení požadovaného objektu v NP.



Obrázek 1 - UC diagram metamodeláře

4. Data flow diagram

Data flow diagram (diagram datových kroků) znázorňuje tok dat mezi jednotlivými funkcemi aplikace. Popisuje funkce a jejich vazby. V diagramu níže je, mimo jiné, znázorněn tok dat mezi MMN a NP. Tento tok nastává v případě, kdy NP potřebuje vykreslit nějaký objekt. NP požádá MMN o vytvoření objektu a po následném obdržení požadovaného objektu je objekt v plné režii NP. Na druhou stranu MMN generuje XML popisující modelovanou notaci a posílá ji na výstup (ukládá na zvolené datové uložště).



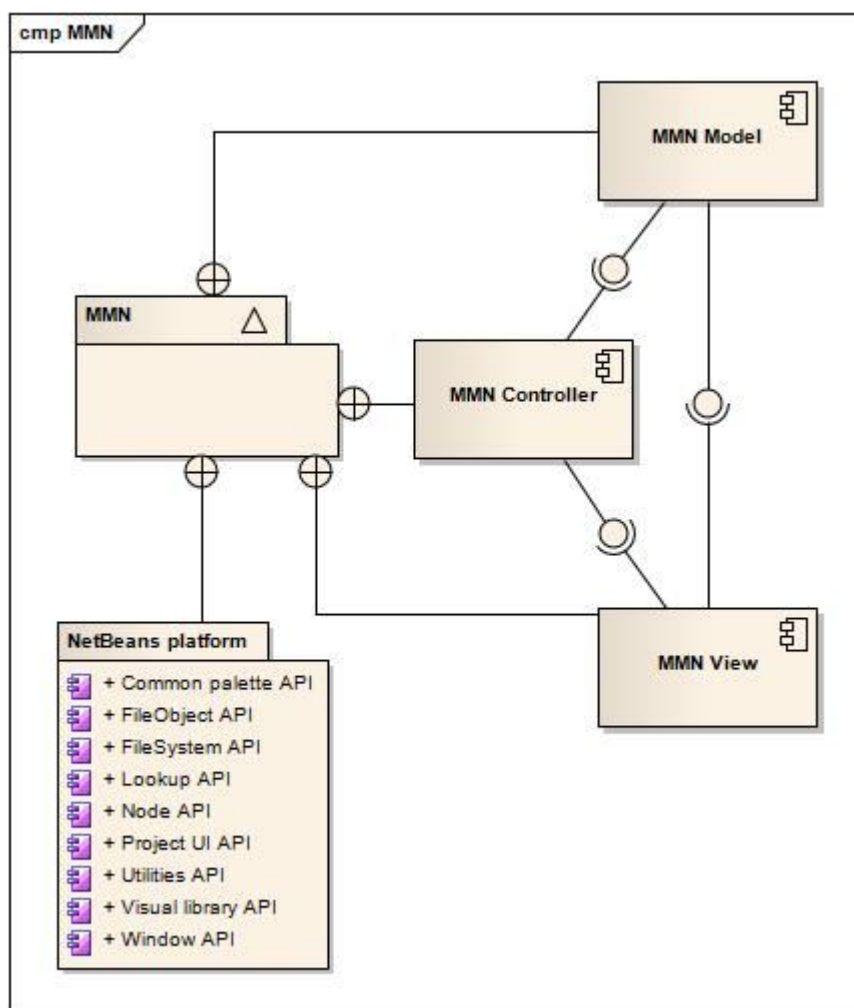
Obrázek 2 - Data flow diagram

5. Class diagram

Class diagram znázorňuje základní stavební prvky aplikace a jejich vazby.

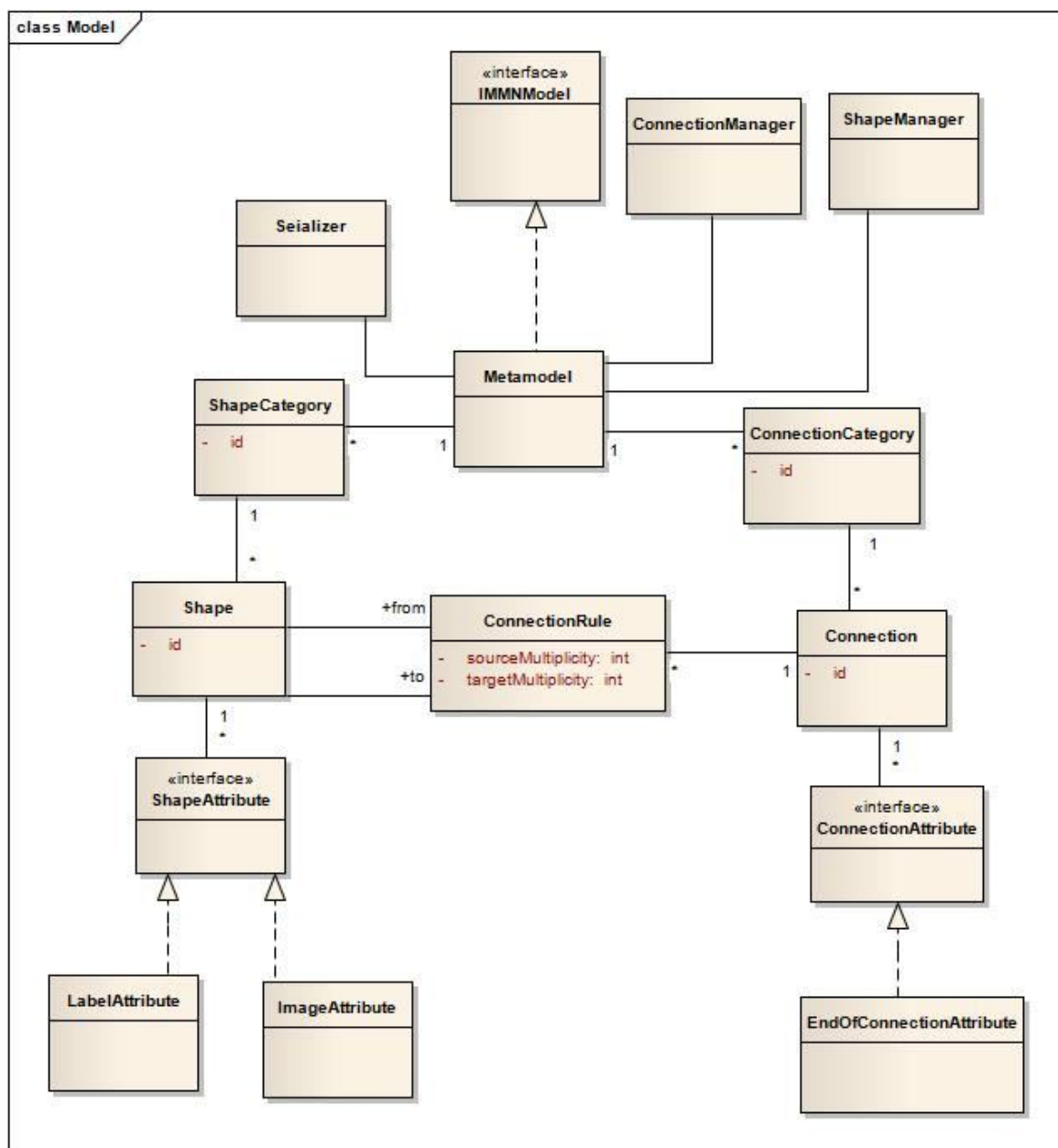
Náhled na analytický model

Na obrázku 3 je znázorněné postavení MMN vůči Netbeans platformě a jsou zde také znázorněné jednotlivé části MMN a jejich závislosti. Jak je vidět z obrázku, MMN je postavené na MVC architektonickém vzoru.



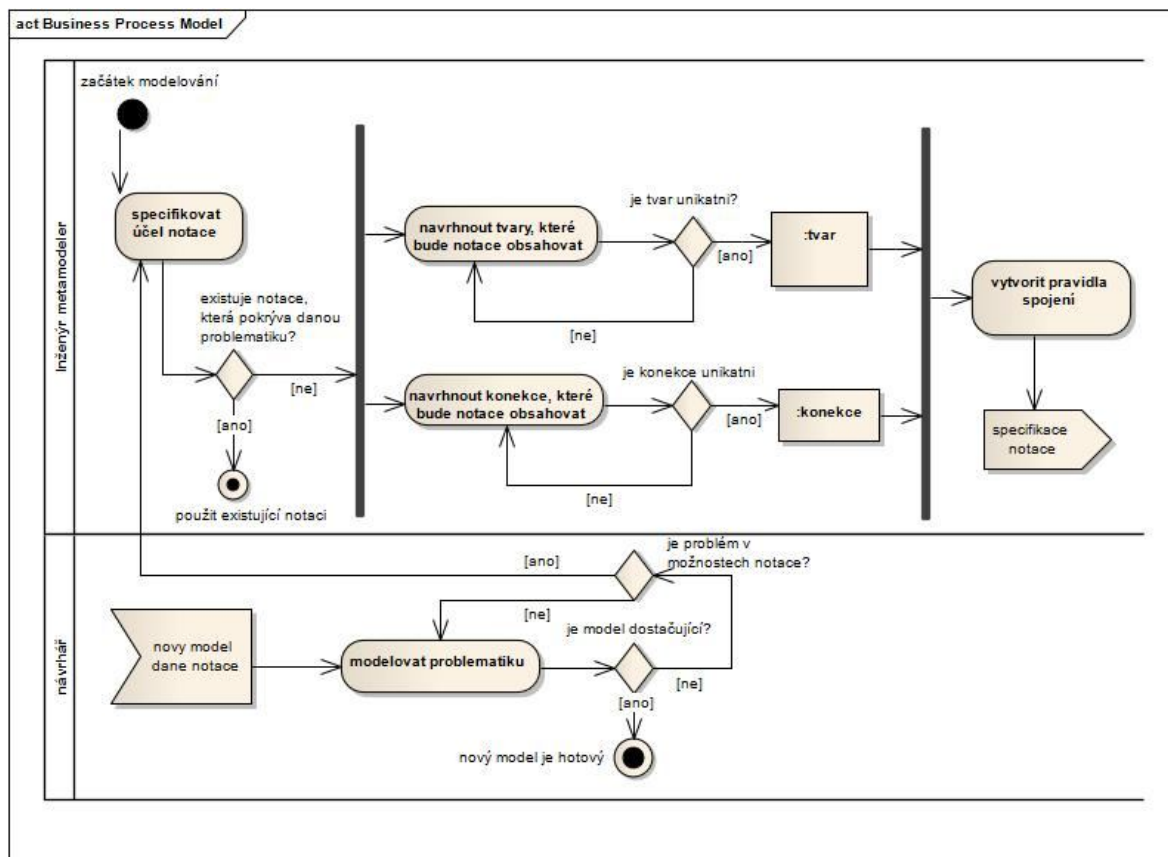
Obrázek 3 - Postavení MMN vůči NetBeans platformě

Class diagram Model komponenty



Obrázek 4 - Class diagram MMN View komponenty

6. Diagram aktivit



Obrázek 5 - Diagram aktivit