

# Komponenta Communicator

Technický popis

a

Návod k obsluze

Pavel Štrof, Jan Šembera a kol.

DHI, a.s.

Technická univerzita v Liberci

Závěrečná zpráva pro projekt MOKOTRAN

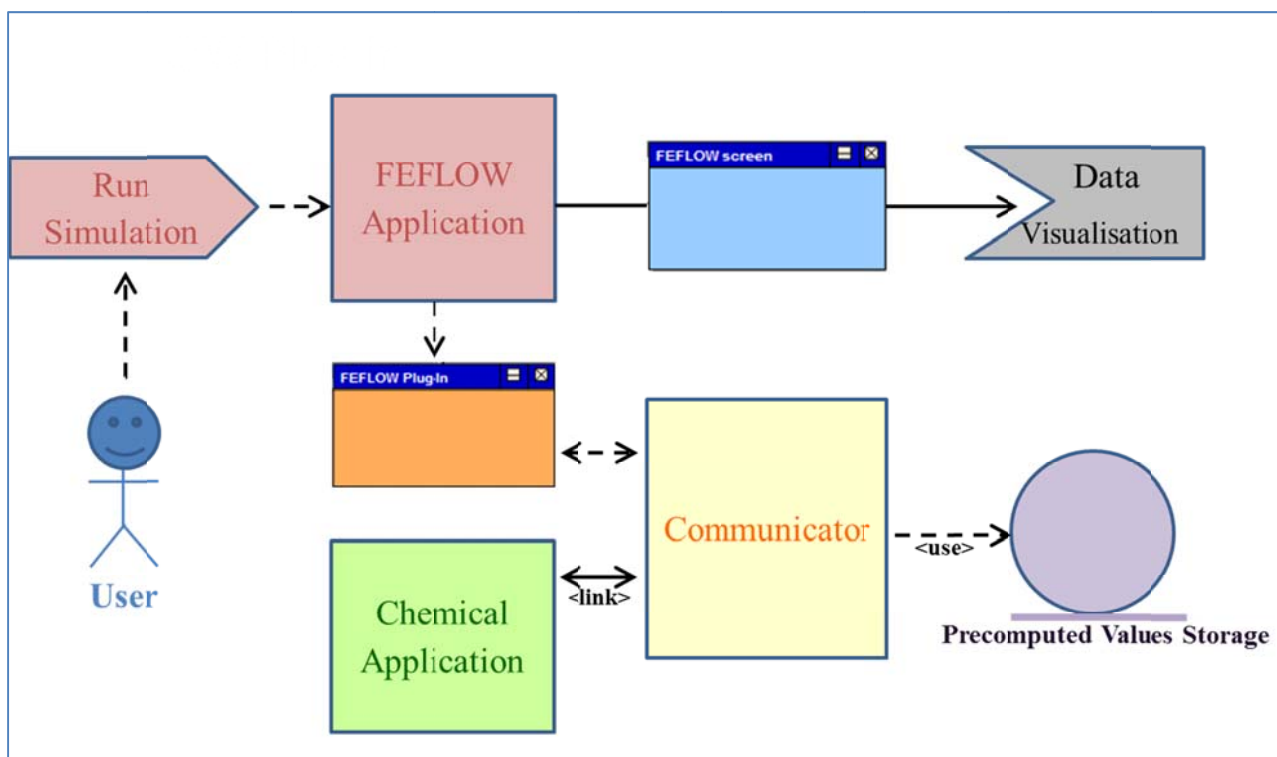
**prosinec 2015**

## Úvod

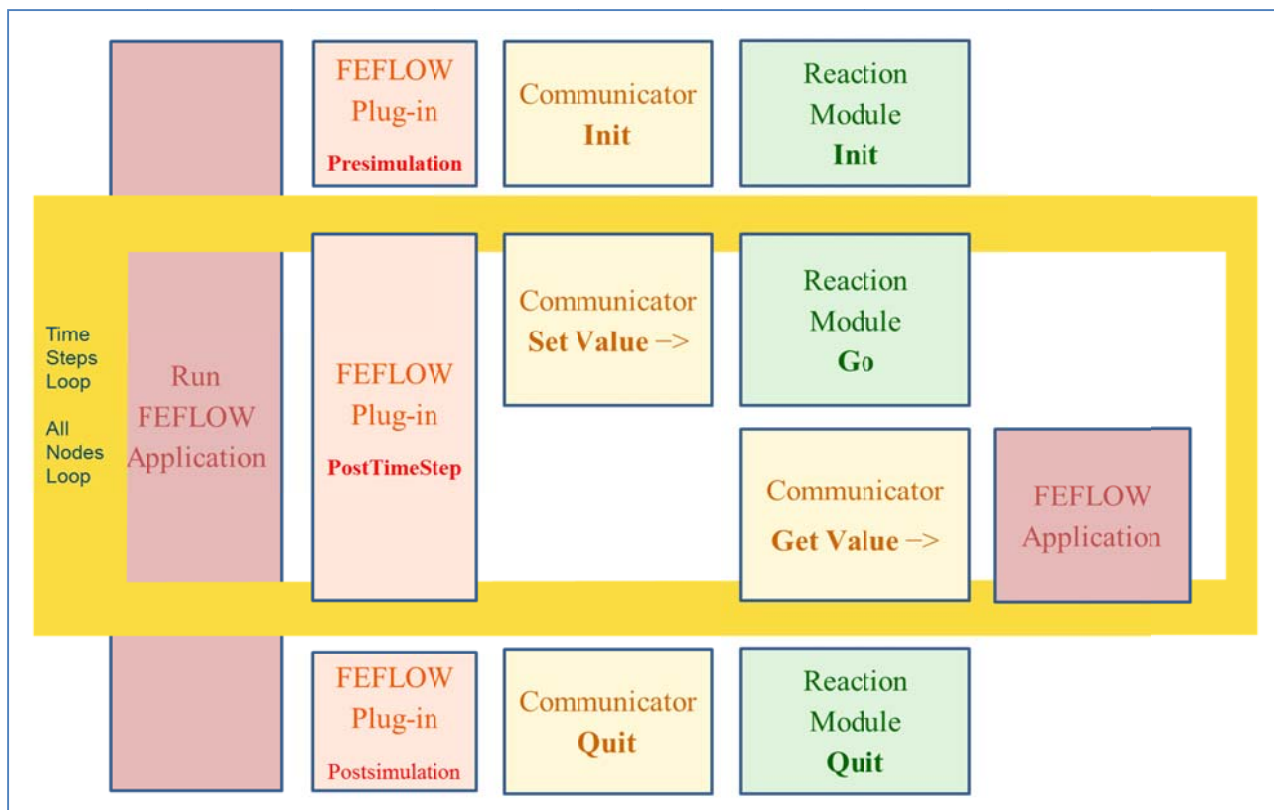
Tato zpráva shrnuje hlavní výsledek 1. aktivity projektu Mokotran za poslední rok řešení. Na základě předchozího řešení projektu a výsledku z roku 2015 byl zpracován Technický popis komponenty Communicator uvedený v následující kapitole, podrobnější informace o programové realizaci je možno nalézt v příloze **MOCO Technical Solution** (anglický jazyk). Příklad s návodem k použití je zaměřen na popis konfiguračních souborů včetně příkladu, který odpovídá situaci řešené v rámci Aktivit 2 a 3.

## Technický popis

Komunikační rozhraní mezi transportním a reakčním modelem je koncipováno v trojvrstvé architektuře. Prostřední vrstva (Communicator) má za úkol zpracování vlastní logiky přenosu dat o transportovaných chemických látkách, jejich ukládání v databázi předpočítaných hodnot a vyhodnocení podmínek jejich dalšího využití v kombinaci s voláním reakčního modelu. První vrstva přebírá data z transportního modelu a vrací hodnoty po volání reakčního modelu, třetí vrstva řeší komunikaci s modelem reakčním, předání vstupních hodnot pro výpočet a načtení výstupních hodnot reakčního modelu. Bylo vytvořeno řešení pro kombinaci transportního modelu FEFLOW a reakčního modelu Geochemist's Workbench (GWB). Schéma práce tohoto doplňku pro FEFLOW je na obrázku:



Komponenta Communicator pracuje se třemi vnitřními funkcionalitami. První řeší inicializaci reakčního modelu GWB podle složek transportovaných v modelu FEFLOW, druhá má za úkol v každém časovém kroku a v každém uzlu FEM sítě volání modelu GWB a třetí uzavírá komunikační kanály po skončení simulace FEFLOW modelu.



Řešení je založeno na algoritmu, kdy v každém časovém řádku a v něm v každém elementu (ve FEFLOW v uzlu) prostorové sítě je volán reakční model s koncentracemi složek pevné i kapalně fáze, které jsou výsledkem míchání řešeného transportním modelem. Současně je vypočtená situace ukládána do databáze a pomocí SQL příkazů definovaných v databázovém schématu je vyhodnocováno v následujících časových řádcích, jestli není možné výpočet nahradit výsledkem z databáze. Schéma datového modelu MOCO realizovaného v prostředí PostgreSQL je v příloze **MOCO Technical Solution**. Algoritmus podobnosti je realizován pomocí SQL příkazů uložených v podadresáři SQL ve složce, kde je instalována komponenta Communicator. Je tak umožněna jeho uživatelská formulace bez nutnosti programátorských zásahů do instalovaných knihoven DLL. Do tabulky *moco.input\_component* jsou ukládány hodnoty koncentrací složek posílaných do reakčního modelu. V tabulce *moco.output\_component* jsou k dispozici výstupy z reakčního modelu. Procedura *moco.f\_get\_rank* počítá normu vzdálenosti vektoru hodnot koncentrací složek od referenčního bodu. Novému vektoru hodnot je vypočtena norma vzdálenosti a porovnána s normami předchozích vektorů uložených v databázi. Jestliže tato nová hodnota je dostatečně blízko některé z předchozích, tak místo volání výpočtu reakčního modelu jsou do transportního modelu vráceny hodnoty z databáze. Funkce *moco.f\_search\_outputs* hledá v příkladu vzdálenost *diff* definovanou jako kosočtvercová norma, editací souboru *moco.SQL* v podadresáři SQL je možné definovat jinou normu (Euklidova, čtvercová, ...) podle potřeb uživatele. Úplný výpis SQL příkazů je uveden v příloze **MOCO Technical Solution**. Celá instalace komponenty, vzorových XML souborů a SQL příkazů potřebných ke generování datového úložiště je k dispozici ke stažení na stránkách projektu MOKOTRAN.

Komunikace uvnitř vrstev komponenty je konfigurovatelné pomocí XML souborů, popis odpovídajících šablon XSD je předmětem následujícího textu.

## XSD šablona vrstvy ReactionModuleGWB

attribute form default: **unqualified**

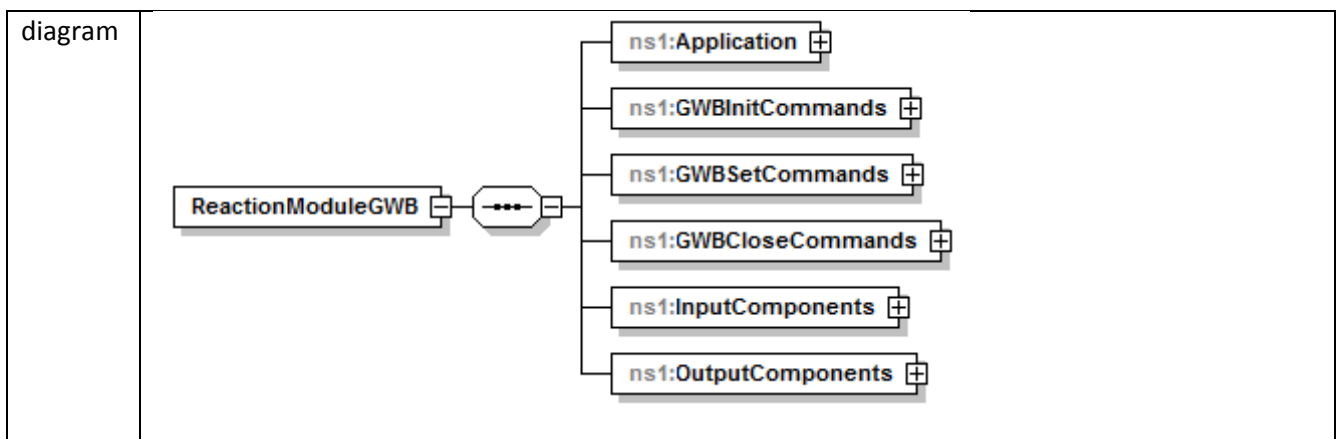
element form default: **qualified**

targetNamespace: **DHI.ContaminantsMobility**

Elements

### [ReactionModuleGWB](#)

element **ReactionModuleGWB**



element **ReactionModuleGWB/Application**



attribute **ReactionModuleGWB/Application/@name**

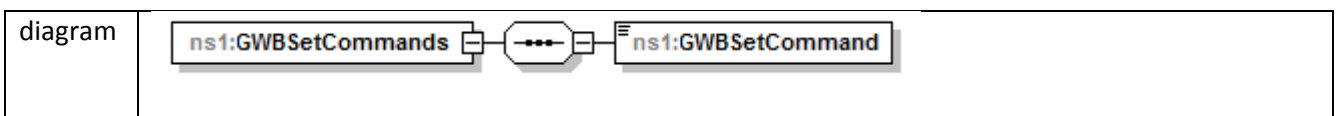
element **ReactionModuleGWB/GWBInitCommands**



element **ReactionModuleGWB/GWBInitCommands/GWBInitCommand**



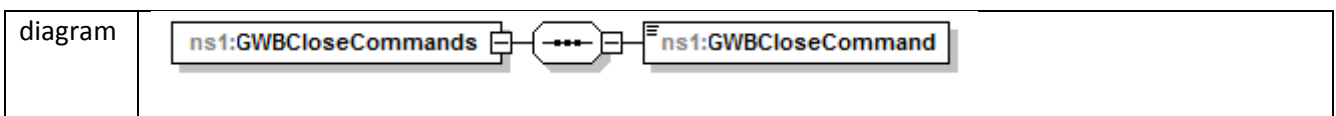
element **ReactionModuleGWB/GWBSetCommands**



element **ReactionModuleGWB/GWBSetCommands/GWBSetCommand**



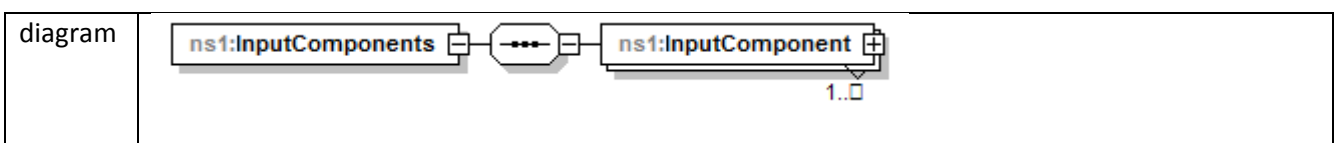
element **ReactionModuleGWB/GWBCloseCommands**



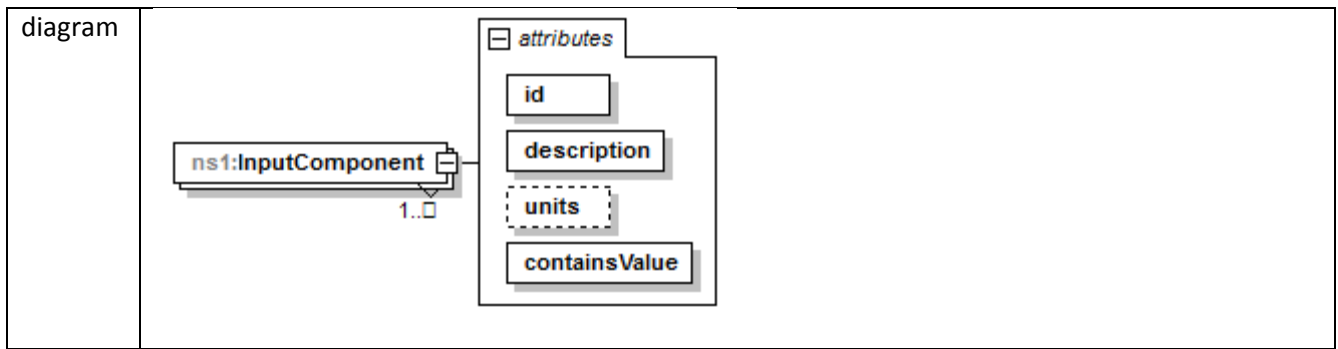
element **ReactionModuleGWB/GWBCloseCommands/GWBCloseCommand**



element **ReactionModuleGWB/InputComponents**



element **ReactionModuleGWB/InputComponents/InputComponent**



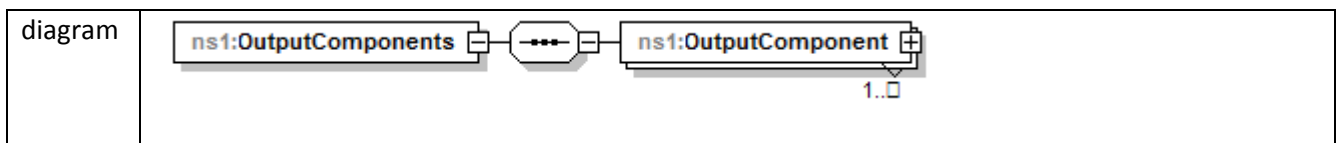
attribute **ReactionModuleGWB/InputComponents/InputComponent/@id**

attribute **ReactionModuleGWB/InputComponents/InputComponent/@description**

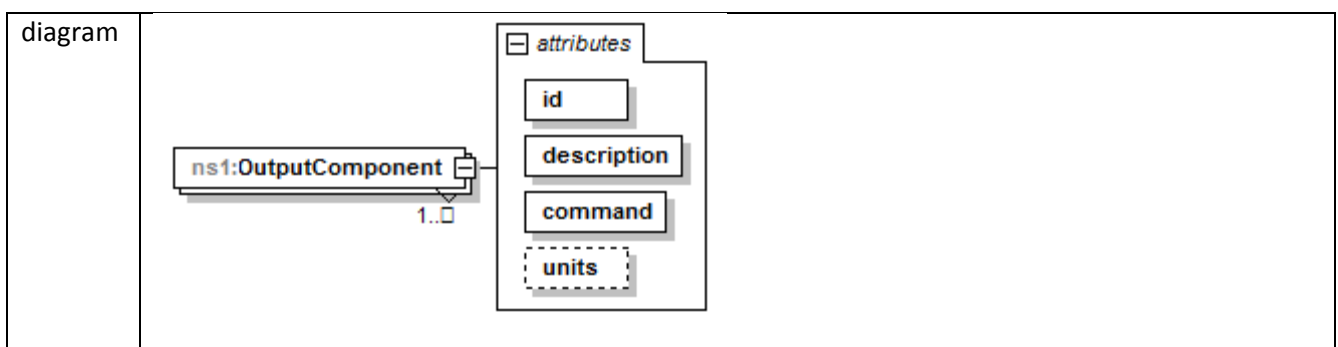
attribute **ReactionModuleGWB/InputComponents/InputComponent/@units**

attribute **ReactionModuleGWB/InputComponents/InputComponent/@containsValue**

element **ReactionModuleGWB/OutputComponents**



element **ReactionModuleGWB/OutputComponents/OutputComponent**



attribute **ReactionModuleGWB/OutputComponents/OutputComponent/@id**

attribute **ReactionModuleGWB/OutputComponents/OutputComponent/@description**

attribute **ReactionModuleGWB/OutputComponents/OutputComponent/@command**

attribute **ReactionModuleGWB/OutputComponents/OutputComponent/@units**

## TransportModuleFEFLOW

attribute form default: **unqualified**

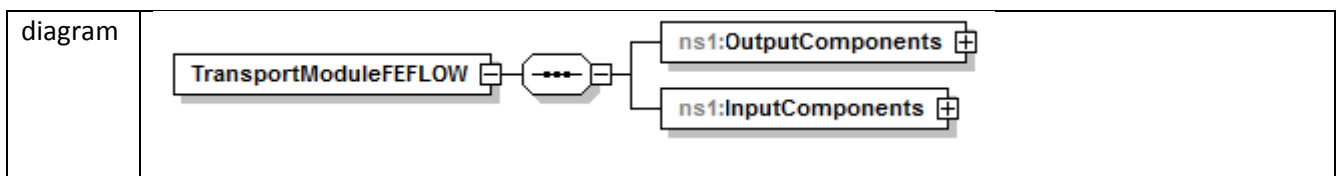
element form default: **qualified**

targetNamespace: **DHI.ContaminantsMobility**

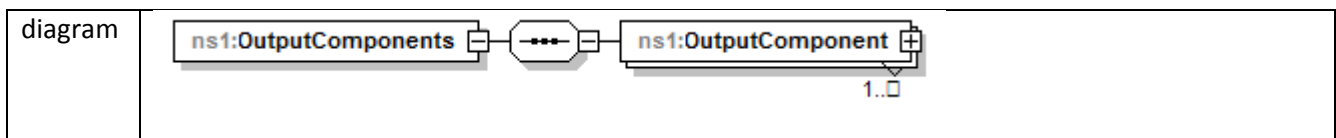
Elements

### [TransportModuleFEFLOW](#)

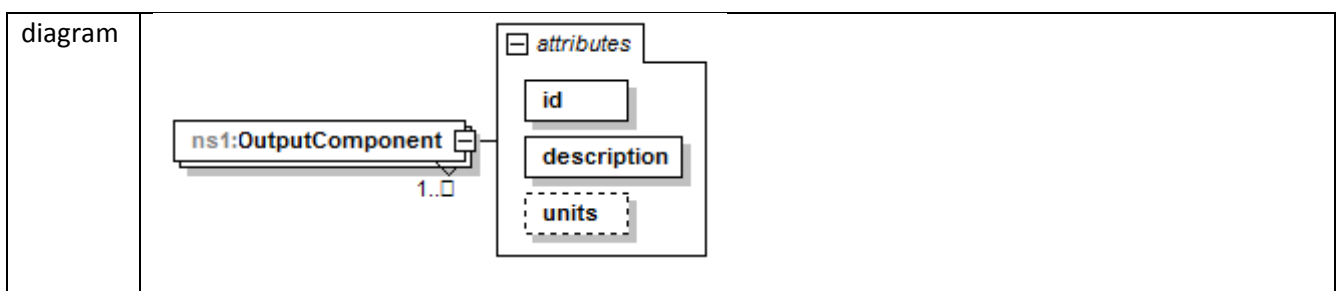
element **TransportModuleFEFLOW**



element **TransportModuleFEFLOW/OutputComponents**



element **TransportModuleFEFLOW/OutputComponents/OutputComponent**



attribute **TransportModuleFEFLOW/OutputComponents/OutputComponent/@id**

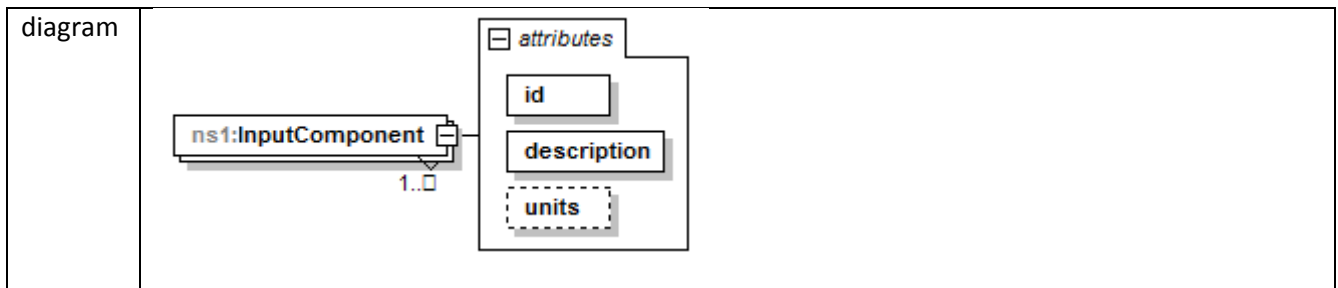
attribute **TransportModuleFEFLOW/OutputComponents/OutputComponent/@description**

attribute **TransportModuleFEFLOW/OutputComponents/OutputComponent/@units**

element **TransportModuleFEFLOW/InputComponents**



element **TransportModuleFEFLOW/InputComponents/InputComponent**



attribute **TransportModuleFEFLOW/InputComponents/InputComponent/@id**

attribute **TransportModuleFEFLOW/InputComponents/InputComponent/@description**

attribute **TransportModuleFEFLOW/InputComponents/InputComponent/@units**

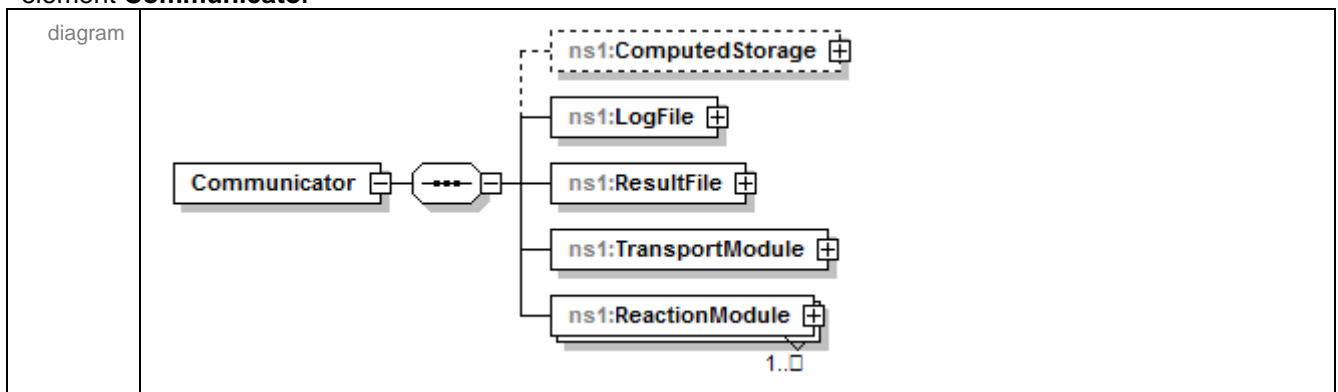


## XSD šablona vrstvy Communicator

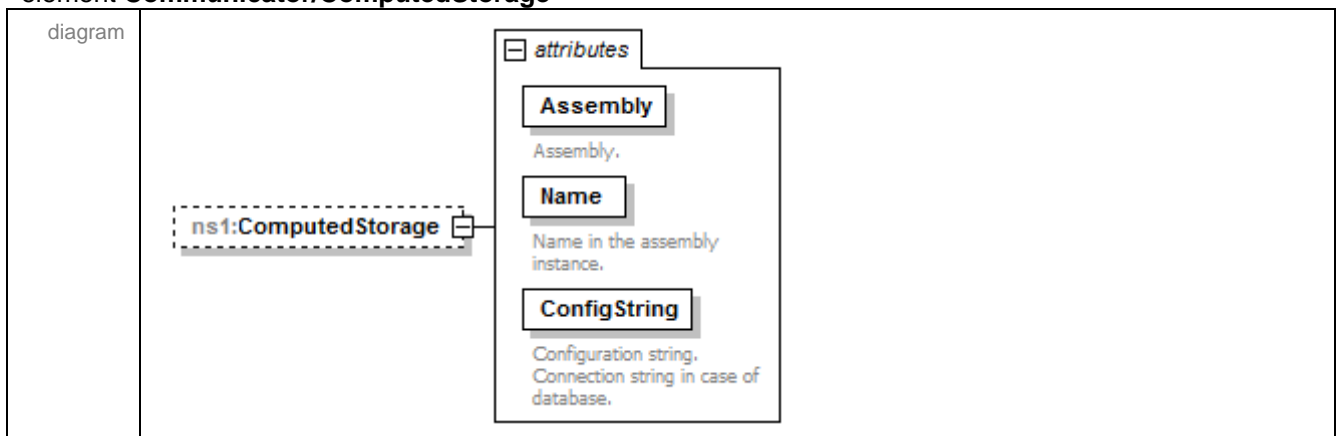
attribute form default: **unqualified**  
 element form default: **qualified**  
 targetNamespace: **DHI.ContaminantsMobility**

Elements  
[Communicator](#)

### element Communicator



### element Communicator/ComputedStorage

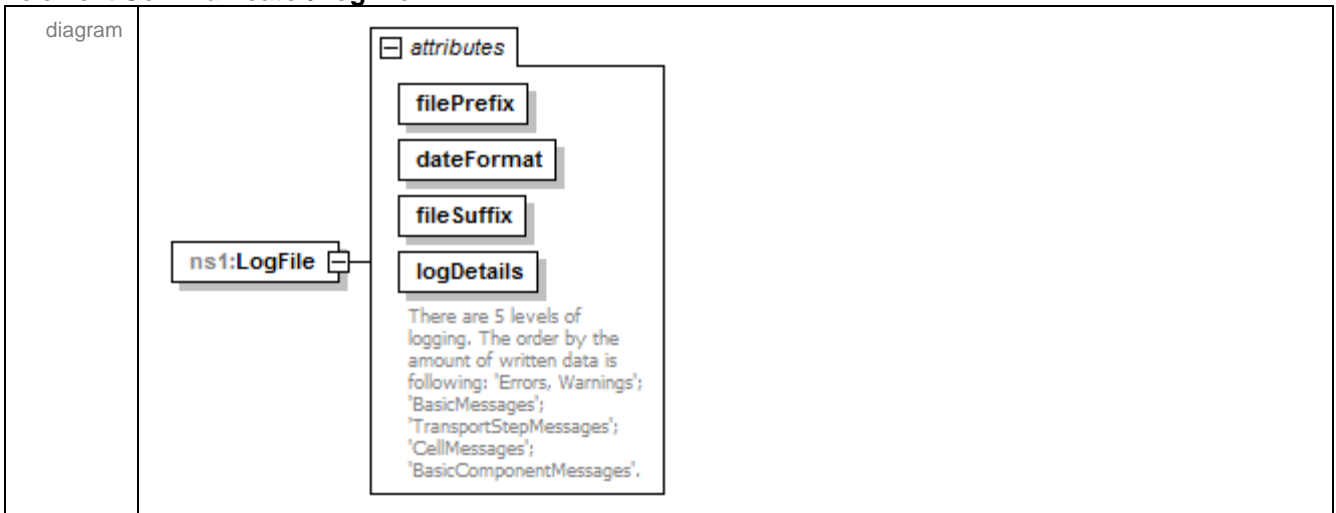


attribute **Communicator/ComputedStorage/@Assembly**

attribute **Communicator/ComputedStorage/@Name**

attribute **Communicator/ComputedStorage/@ConfigString**

element **Communicator/LogFile**



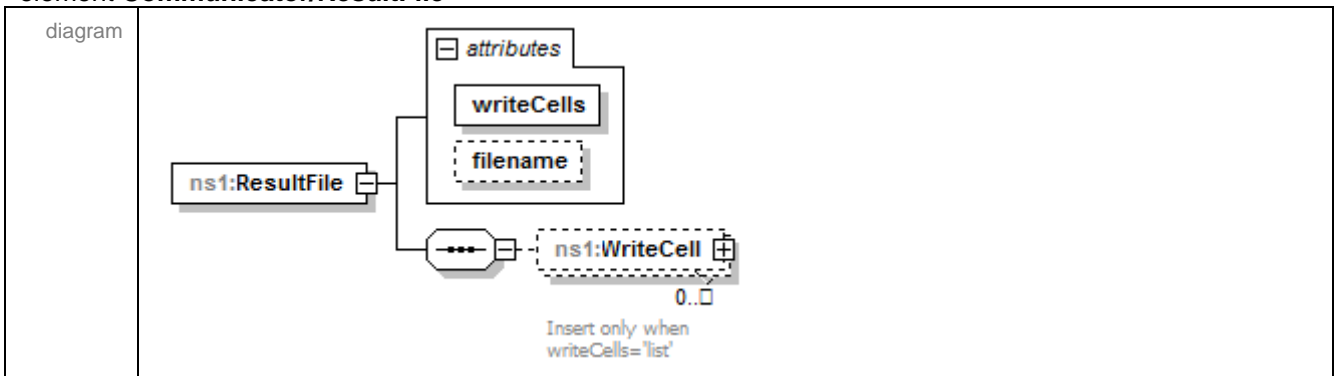
attribute **Communicator/LogFile/@filePrefix**

attribute **Communicator/LogFile/@dateFormat**

attribute **Communicator/LogFile/@fileSuffix**

attribute **Communicator/LogFile/@logDetails**

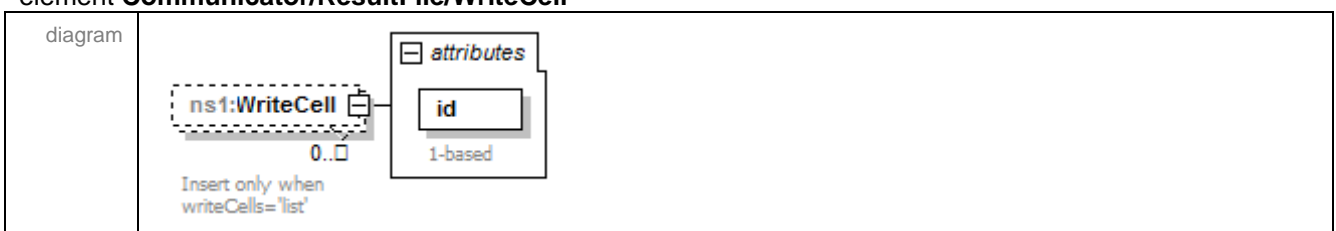
element **Communicator/ResultFile**



attribute **Communicator/ResultFile/@writeCells**

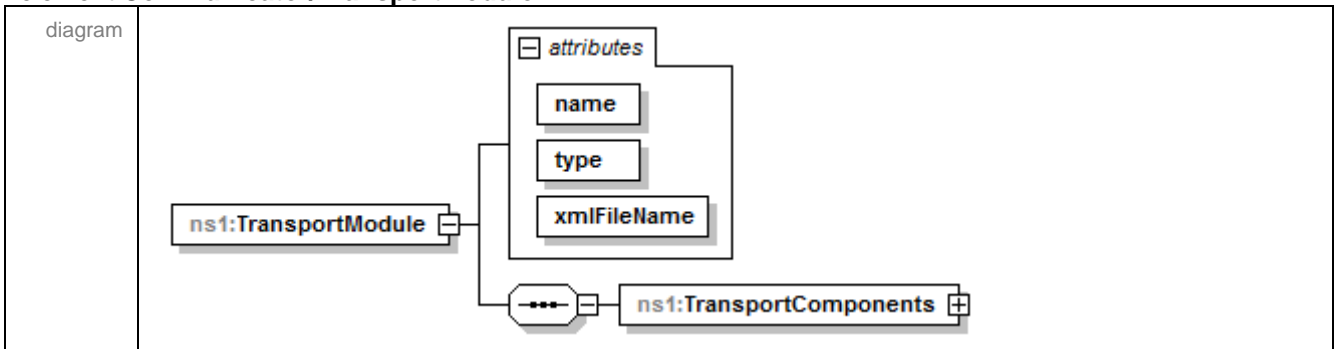
attribute **Communicator/ResultFile/@filename**

element **Communicator/ResultFile/WriteCell**



attribute **Communicator/ResultFile/WriteCell/@id**

element **Communicator/TransportModule**

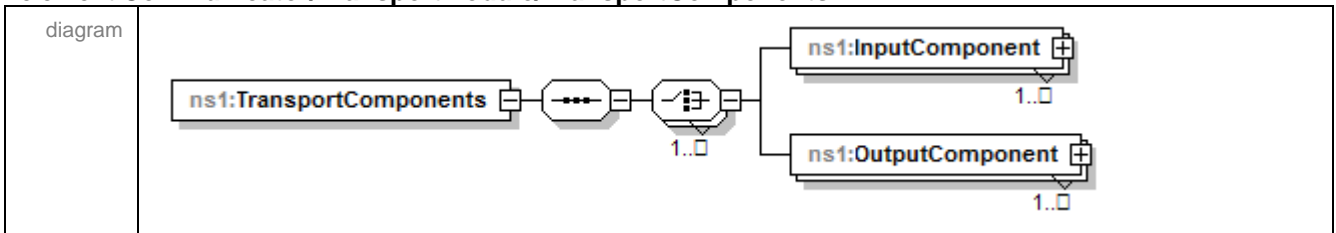


attribute **Communicator/TransportModule/@name**

attribute **Communicator/TransportModule/@type**

attribute **Communicator/TransportModule/@xmlFileName**

element **Communicator/TransportModule/TransportComponents**



element **Communicator/TransportModule/TransportComponents/InputComponent**



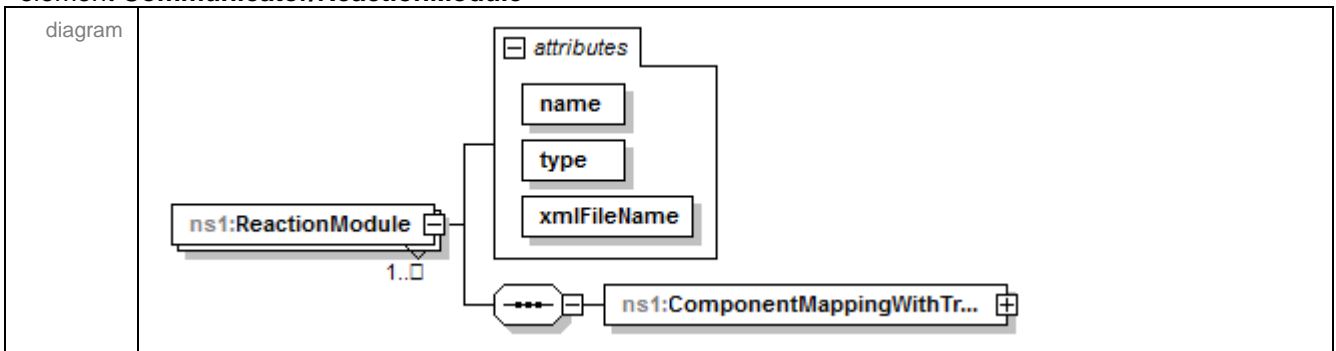
attribute **Communicator/TransportModule/TransportComponents/InputComponent/@id**

element **Communicator/TransportModule/TransportComponents/OutputComponent**



attribute **Communicator/TransportModule/TransportComponents/OutputComponent/@id**

element **Communicator/ReactionModule**

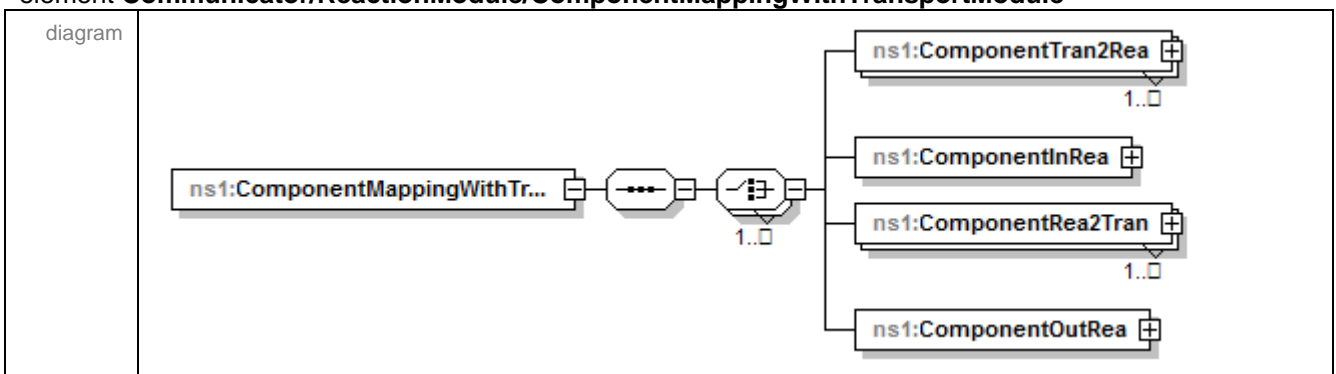


attribute **Communicator/ReactionModule/@name**

attribute **Communicator/ReactionModule/@type**

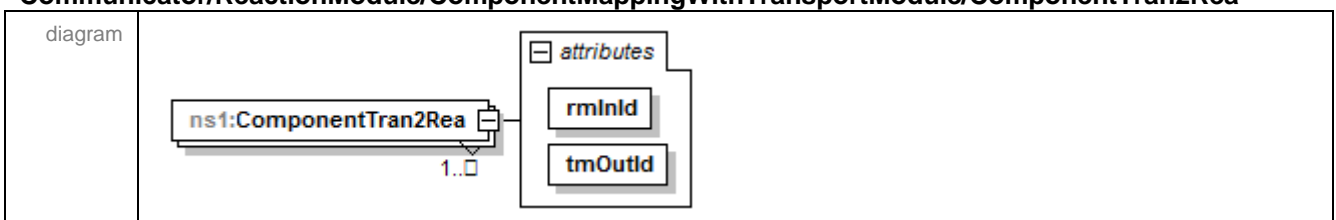
attribute **Communicator/ReactionModule/@xmlFileName**

element **Communicator/ReactionModule/ComponentMappingWithTransportModule**



element

**Communicator/ReactionModule/ComponentMappingWithTransportModule/ComponentTran2Rea**



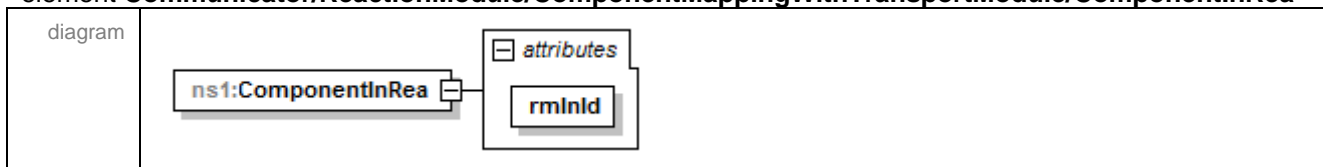
attribute

**Communicator/ReactionModule/ComponentMappingWithTransportModule/ComponentTran2Rea/@rmlnId**

attribute

**Communicator/ReactionModule/ComponentMappingWithTransportModule/ComponentTran2Rea/@tmOutId**

element **Communicator/ReactionModule/ComponentMappingWithTransportModule/ComponentInRea**

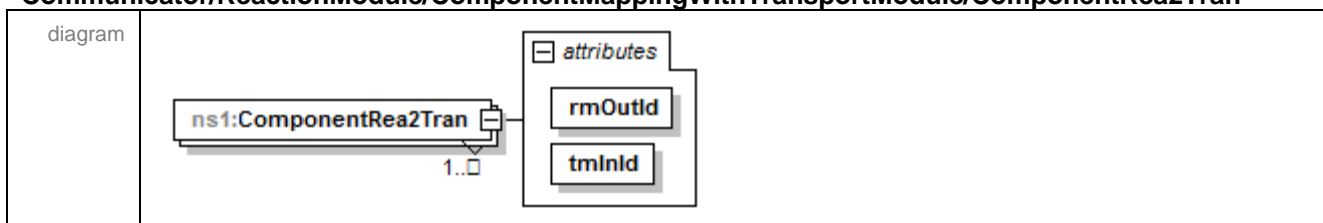


attribute

**Communicator/ReactionModule/ComponentMappingWithTransportModule/ComponentInRea/@rmInId**

element

**Communicator/ReactionModule/ComponentMappingWithTransportModule/ComponentRea2Tran**



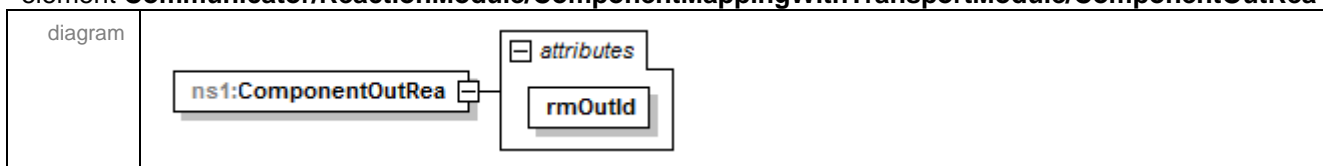
attribute

**Communicator/ReactionModule/ComponentMappingWithTransportModule/ComponentRea2Tran/@rmOutId**

attribute

**Communicator/ReactionModule/ComponentMappingWithTransportModule/ComponentRea2Tran/@tmlInId**

element **Communicator/ReactionModule/ComponentMappingWithTransportModule/ComponentOutRea**



attribute

**Communicator/ReactionModule/ComponentMappingWithTransportModule/ComponentOutRea/@rmOutId**

## Návod k obsluze

Pro správný běh komunikace mezi transportním a reakčním modelem je nutné zavést indexy odpovídajících složek obou systémů. Tento proces byl realizován pomocí samostatných XML konfiguračních souborů podle šablon podrobně popsanych v níže uvedených tabulkách, jejich technický popis je uveden v předcházející kapitole. Jak již jejich název napovídá, tak uvedený příklad odpovídá transportnímu modelu v prostředí FEFLOW a reakčnímu modelu v GWB. Uvedený postup ale je možno realizovat i pro jiná prostředí – např. MODFLOW a PHREEQ. Konfigurační soubory jsou dále využity pro definici příkazů spouštění běhu reakčního modelu, jeho inicializaci, tak aby byla příslušně instance zavedena v paměťovém prostoru dostupná po celou dobu běhu modelu transportního, a jeho ukončení souvisejícím s uvolněním paměťového prostoru na konci celé simulace.

<b>ReactionModuleGWB.xsd</b>	GWBInitCommands	Inicializační příkazy			
	GWBSetCommands	Příkaz běhu			
	GWBCloseCommands	Příkazy ukončení			
	InputComponent	id	description	units	containsValue
		<i>Zavedení indexů složek vstupujících do reakčního modelu</i>	<i>Příkaz/popis</i>	<i>jednotky</i>	<i>true or false</i>
	OutputComponent	id	description	command	units
<i>Zavedení indexů složek vystupujících z reakčního modelu</i>		<i>Popis</i>	<i>Příkaz</i>	<i>jednotky</i>	

<b>TransportModuleFEFLOW.xsd</b>	OutputComponent	id	description	units	
		<i>Zavedení indexů složek vystupujících z transportního modelu</i>	<i>Jméno složky</i>	<i>jednotky</i>	
	InputComponent	id	description		
		<i>Zavedení indexů složek vracených zpět po výpočtu reakce</i>	<i>Jméno složky</i>		

<b>Communicator.xsd</b>	OutputComponent	id	
		<i>Seznam indexů složek vystupujících z transportního modelu</i>	
	InputComponent	id	
		<i>Seznam indexů složek vrácených zpět po výpočtu reakce</i>	
	ComponentTran2Rea	rmInId	tmOutId
		<i>Mapování složek transportního modelu <b>tmOutId</b> na složky modelu reakčního <b>rmInId</b></i>	
	ComponentInRea	rmInId	
		<i>Virtuální složky, nepředávají se hodnoty, umožňuje zadávat podmínky výpočtu</i>	
	ComponentRea2Tran	rmOutId	tmlnId
		<i>Mapování složek reakčního modelu <b>rmOutId</b> na složky modelu transportního <b>tmlnId</b></i>	
ComponentOutRea			
	<i>Vypočtené hodnoty z reakčního modelu, které je užitečné zobrazovat v modelu transportním</i>		

Příklad XML souborů pro složení vody (Na+, K+, Ca++, Mg++, Fe++, Mn++, Cl, SO4--, HCO3-, UO2++) je v následujících tabulkách. Výpočet pH a Eh je exportován zpět do prostředí FEFLOW a je tak umožněno pracovat s časovým i prostorovým rozložením těchto hodnot ve standardním prostředí FEFLOW GUI. Teplota roztoku je do reakčního výpočtu exportována z prostředí FEFLOW, s výhodou je využita i možnost simulace rozložení teplotního pole pomocí jeho nástrojů. Hodnoty pH jsou v příkladu počítány podle elektroneutality dopočítávané na bilanci H<sup>+</sup>, Eh je v příkladu řešeno výpočtem založeným na bilanci O<sub>2</sub>. Konfigurační mechanismus umožňuje i zadávání podmínek potlačení rovnováh vůči vybraným minerálům (v GWB příkaz suppress). Komunikaci mezi komponentou a GWB lze sledovat v log souboru. Detail výpisů lze nastavovat v atributu *logDetails* elementu <LogFile> ve schématu Communicator.xsd v pěti možných úrovních

- Errors, Warnings
- BasicMessages
- TransportStepMessages
- CellMessages
- BasicComponentMessages

Příklad výpisu log file:

```
20:55:35 : Communicator> Communicator> Logging started.
20:55:35 : Communicator> Communicator> Result file writing initialized with writeCells = 'list'.
20:55:35 : Communicator> Communicator> Reading configuration of transport module.
20:55:35 : TransportModuleFEFLOW> Init> Successful.
20:55:35 : TransportModuleFEFLOW> ProvidedComponents> Successful.
20:55:35 : TransportModuleFEFLOW> RequiredComponents> Successful.
20:55:35 : Communicator> Communicator> Transport module loaded successfully.
20:55:35 : Communicator> Communicator> Reading configuration of reaction module 0.
20:55:36 : ReactionModuleGWB> Init> Opening GWB application through a pipe with result 1.
20:55:36 :   Sending command> reset
20:55:36 :   Sending command> delxi = 1 linear
20:55:36 :   Sending command> print off
20:55:36 :   Sending command> plot off
20:55:36 : ReactionModuleGWB> Init> Successful.
20:55:36 : ReactionModuleGWB> ProvidedComponents> Successful.
20:55:36 : ReactionModuleGWB> RequiredComponents> Successful.
20:55:36 : Communicator> Communicator> Reaction module 0 loaded successfully.
20:55:36 : Communicator> Communicator> Successful.
20:55:37 : Communicator> DoExchange> Time step index: 1
20:55:37 : Communicator> DoExchange> Node index: 1
20:55:37 : TransportModuleFEFLOW> GetValues> Successful.
20:55:37 :   Sending command> time start = 0 day, end = 1E-10 day
20:55:37 :   Sending command> temperature = 13.2 C
20:55:37 :   Sending command> H2O = 1 free kg
20:55:37 :   Sending command> balance on H+
20:55:37 :   Sending command> suppress Dolomite Dolomite-dis Dolomite-ord Ferrite-Ca Ferrite-Cu Ferrite-Dicalcium
Ferrite-Mg Ferrite-Zn Goethite Hematite Magnetite Siderite UO2.25 UO2.25(beta) UO2.3333(beta)
20:55:37 :   Sending command> Na+ = 8.2 mg/l
20:55:37 :   Sending command> K+ = 1.4 mg/l
20:55:37 :   Sending command> Ca++ = 23.5 mg/l
20:55:37 :   Sending command> Mg++ = 5.8 mg/l
20:55:37 :   Sending command> Fe++ = 1E-11 mg/l
20:55:37 :   Sending command> Mn++ = 1E-10 mg/l
20:55:37 :   Sending command> Cl- = 5.34 mg/l
20:55:37 :   Sending command> SO4-- = 60.2 mg/l
20:55:37 :   Sending command> HCO3- = 98 mg/l
20:55:37 :   Sending command> UO2++ = 0.001 mg/l
20:55:37 :   Sending command> O2(aq) = 1.278E-12 mg/l
20:55:37 :   Sending command> go
20:55:37 : ReactionModuleGWB> SetValue> Successful.
20:55:37 :   Requesting result> concentration Na+; units> mg/l
20:55:37 :   Response returned> 8.20000000000001
20:55:37 :   Requesting result> concentration K+; units> mg/l
20:55:37 :   Response returned> 1.4
20:55:37 :   Requesting result> concentration Ca++; units> mg/l
20:55:37 :   Response returned> 23.5000000000004
20:55:37 :   Requesting result> concentration Mg++; units> mg/l
20:55:37 :   Response returned> 5.80000000000014
20:55:37 :   Requesting result> concentration Fe++; units> mg/l
20:55:37 :   Response returned> 1.00000000000016E-11
20:55:37 :   Requesting result> concentration Mn++; units> mg/l
20:55:37 :   Response returned> 1.00000000000002E-10
20:55:37 :   Requesting result> concentration Cl-; units> mg/l
20:55:37 :   Response returned> 5.34000000000001
20:55:37 :   Requesting result> concentration SO4--; units> mg/l
```



20:55:37 : Response returned> 60.2000000000014  
20:55:37 : Requesting result> concentration HCO3-; units> mg/l  
20:55:37 : Response returned> 98.0000000000051  
20:55:37 : Requesting result> concentration UO2++; units> mg/l  
20:55:37 : Response returned> 0.0010000000000012  
20:55:37 : Requesting result> concentration O2(aq); units> mg/l  
20:55:37 : Response returned> 1.2780000000022E-12  
20:55:37 : Requesting result> pH; default units  
20:55:37 : Response returned> 6.23053864244329  
20:55:37 : Requesting result> Eh; units> V  
20:55:37 : Response returned> 0.647737338758893  
20:55:37 : ReactionModuleGWB> GetResults> Successful.  
20:55:37 : TransportModuleFEFLOW> SetResults> Successful.  
20:55:37 : Communicator> DoExchange> Node index: 2  
20:55:37 : TransportModuleFEFLOW> GetValues> Successful.  
20:55:37 : Sending command> time start = 0 day, end = 1E-10 day  
20:55:37 : Sending command> temperature = 13.2 C  
20:55:37 : Sending command> H2O = 1 free kg  
20:55:37 : Sending command> balance on H+  
20:55:37 : Sending command> suppress Dolomite Dolomite-dis Dolomite-ord Ferrite-Ca Ferrite-Cu Ferrite-Dicalcium  
Ferrite-Mg Ferrite-Zn Goethite Hematite Magnetite Siderite UO2.25 UO2.25(beta) UO2.3333(beta)  
20:55:37 : Sending command> Na+ = 8.2 mg/l  
20:55:37 : Sending command> K+ = 1.4 mg/l  
20:55:37 : Sending command> Ca++ = 23.5 mg/l  
20:55:37 : Sending command> Mg++ = 5.8 mg/l  
20:55:37 : Sending command> Fe++ = 1E-11 mg/l  
20:55:37 : Sending command> Mn++ = 1E-10 mg/l  
20:55:37 : Sending command> Cl- = 5.34 mg/l  
20:55:37 : Sending command> SO4-- = 60.2 mg/l  
20:55:37 : Sending command> HCO3- = 98 mg/l  
20:55:37 : Sending command> UO2++ = 0.001 mg/l  
20:55:37 : Sending command> O2(aq) = 1.278E-12 mg/l  
20:55:37 : Sending command> go  
20:55:37 : ReactionModuleGWB> SetValue> Successful.  
20:55:37 : Requesting result> concentration Na+; units> mg/l  
20:55:37 : Response returned> 8.2000000000001  
20:55:37 : Requesting result> concentration K+; units> mg/l  
20:55:37 : Response returned> 1.4  
20:55:37 : Requesting result> concentration Ca++; units> mg/l  
20:55:37 : Response returned> 23.5000000000004  
20:55:37 : Requesting result> concentration Mg++; units> mg/l  
20:55:37 : Response returned> 5.80000000000014  
20:55:37 : Requesting result> concentration Fe++; units> mg/l  
20:55:37 : Response returned> 1.0000000000016E-11  
20:55:37 : Requesting result> concentration Mn++; units> mg/l  
20:55:37 : Response returned> 1.0000000000002E-10  
20:55:37 : Requesting result> concentration Cl-; units> mg/l  
20:55:37 : Response returned> 5.3400000000001  
20:55:37 : Requesting result> concentration SO4--; units> mg/l  
20:55:37 : Response returned> 60.2000000000014  
20:55:37 : Requesting result> concentration HCO3-; units> mg/l  
20:55:37 : Response returned> 98.0000000000051  
20:55:37 : Requesting result> concentration UO2++; units> mg/l  
20:55:37 : Response returned> 0.0010000000000012  
20:55:37 : Requesting result> concentration O2(aq); units> mg/l  
20:55:37 : Response returned> 1.2780000000022E-12  
20:55:37 : Requesting result> pH; default units

20:55:37 : *Response returned> 6.23053864244329*  
20:55:37 : *Requesting result> Eh; units> V*  
20:55:37 : *Response returned> 0.647737338758893*  
20:55:37 : *ReactionModuleGWB> GetResults> Successful.*  
20:55:37 : *TransportModuleFEFLOW> SetResults> Successful.*  
. . .  
20:58:18 : *Communicator> ~Communicator> Result file closed if created.*  
20:58:18 : *TransportModuleFEFLOW> Close> Closing TransportModuleFEFLOW.*  
20:58:18 : *TransportModuleFEFLOW> Close> Successful.*  
20:58:18 : *Communicator> ~Communicator> Transport module closed successfully.*  
20:58:18 : *ReactionModuleGWB> Close> Closing GWB application.*  
20:58:18 : *Sending command> quit*  
20:58:18 : *ReactionModuleGWB> Close> Successful.*  
20:58:18 : *Communicator> ~Communicator> Reaction module 0 closed successfully.*  
20:58:18 : *Communicator> ~Communicator> Successful.*

Příklad XML souboru <b>ReactionModuleGWB</b>							
<b>GWBInitCommands</b>				reset delxi = 1 linear print off plot off			
<b>GWBSetCommand</b>				go			
<b>GWBCloseCommand</b>				quit			
<b>InputComponents</b>				<b>OutputComponents</b>			
id	description	units	containsValue	id	description	command	units
101	time start = 0 day, end =	day	true	201	concentration Na+	concentration Na+	mg/l
102	temperature =	C	true	202	concentration K+	concentration K+	mg/l
103	H2O = 1	free kg	false	203	concentration Ca++	concentration Ca++	mg/l
104	balance on H+		false	204	concentration Mg++	concentration Mg++	mg/l
105	suppress .....		false	205	concentration Fe++	concentration Fe++	mg/l
106	Na+ =	mg/l	true	206	concentration Mn++	concentration Mn++	mg/l
107	K+ =	mg/l	true	207	concentration Cl-	concentration Cl-	mg/l
108	Ca++ =	mg/l	true	208	concentration SO4--	concentration SO4--	mg/l
109	Mg++ =	mg/l	true	209	concentration HCO3-	concentration HCO3-	mg/l
110	Fe++ =	mg/l	true	210	concentration UO2++	concentration UO2++	mg/l
111	Mn++ =	mg/l	true	211	concentration O2(aq)	concentration O2(aq)	mg/l
112	Cl- =	mg/l	true	212	pH	pH	
113	SO4-- =	mg/l	true	213	Eh	Eh	V
114	HCO3- =	mg/l	true				
115	UO2++ =	mg/l	true				
116	O2(aq)	mg/l	true				

Příklad XML souboru <b>TransportModuleFEFLOW</b>				
OutputComponents			InputComponents	
Id	description	units	id	description
1001	step endtime	day	2001	Na+
1002	temperature	C	2002	K+
1003	Na+	mg/l	2003	Ca++
1004	K+	mg/l	2004	Mg++
1005	Ca++	mg/l	2005	Fe++
1006	Mg++	mg/l	2006	Mn++
1007	Fe++	mg/l	2007	Cl-
1008	Mn++	mg/l	2008	SO4--
1009	Cl-	mg/l	2009	HCO3-
1010	SO4--	mg/l	2010	UO2++
1011	HCO3-	mg/l	2011	O2(aq)
1012	UO2++	mg/l	2012	pH
1013	O2(aq)	mg/l	2013	Eh

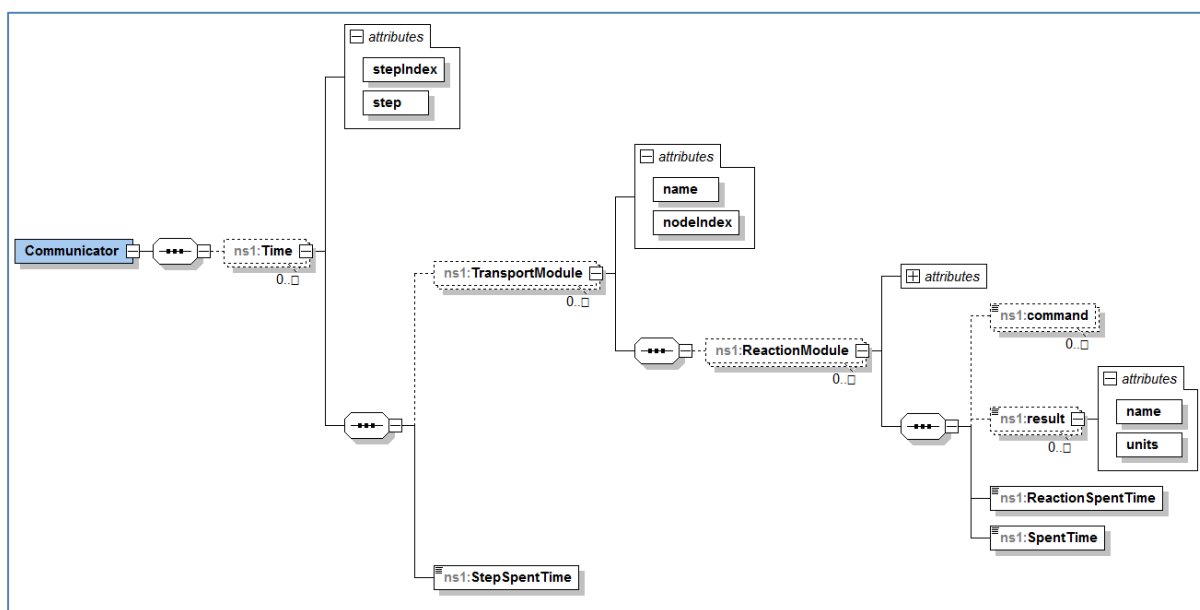
Příklad XML souboru <b>Communicator</b>							
TransportComponents		ComponentMappingWithTransportModule					
InputComponent	OutputComponent	ComponentTran2Rea		ComponentInRea	ComponentRea2Tran		ComponentOutRea
id	id	rmlnId	tmOutId	rmlnId	rmOutId	tmlnId	rmOutId
2001	1001	101	1001	103	201	2001	
2002	1002	102	1002	104	202	2002	
2003	1003	106	1003	105	203	2003	
2004	1004	107	1004		204	2004	
2005	1005	108	1005		205	2005	
2006	1006	109	1006		206	2006	
2007	1007	110	1007		207	2007	
2008	1008	111	1008		208	2008	
2009	1009	112	1009		209	2009	
2010	1010	113	1010		210	2010	
2011	1011	114	1011		211	2011	
2012	1012	115	1012		212	2012	
2013	1013	116	1013		213	2013	

V XML souboru Communicator.xml lze v elementu <ResultFile> řídit výstup do pomocného výstupního souboru mezivýsledků Results.xml. Je možno zde nastavit, z kterých elementů je tento mezivýstup ukládán a v jaké podobě.

Pro analýzu citlivosti propojení FEFLOW – GWB jsou v Results.xml k dispozici elementy <ReactionSpentTime>, <SpentTime> a <StepSpentTime>. Všechny ukládají čas v sekundách. Prvý obsahuje údaj o době, na jak dlouho bylo v daném FEFLOW uzlu a v daném časovém řádku předáno řízení procesu do GWB, je tak uchována časová hodnota, která má znázorňovat vlastní dobu výpočtu simulace GWB pro jednu danou konfiguraci vstupních koncentrací a požadované doby kinetické simulace (typicky délka časového kroku transportu ve FEFLOW).

Element <SpentTime> ukládá údaj o době spotřebované na provádění vlastních algoritmů Communicatoru, tj. rozhodování o nahrazení výpočtu načtením údajů z databáze předpočítaných hodnot včetně případných databázových operací vyhledávání, čtení a ukládání. Na rozdíl od předcházejícího údaje je zde tak uchována i doba, kterou simulační proces strávil ve výpočtu interakce (v našem případě GWB nebo práce s databází).

Element <StepSpentTime> ukládá údaj o době, na jak dlouho předal FEFLOW řízení procesu do komponenty Communicator, tj. včetně dob spotřebované na konverzi dat mezi potřebnými formáty a doba potřebná k přenosu dat z FEFLOW do GWB (databáze) a zpět.



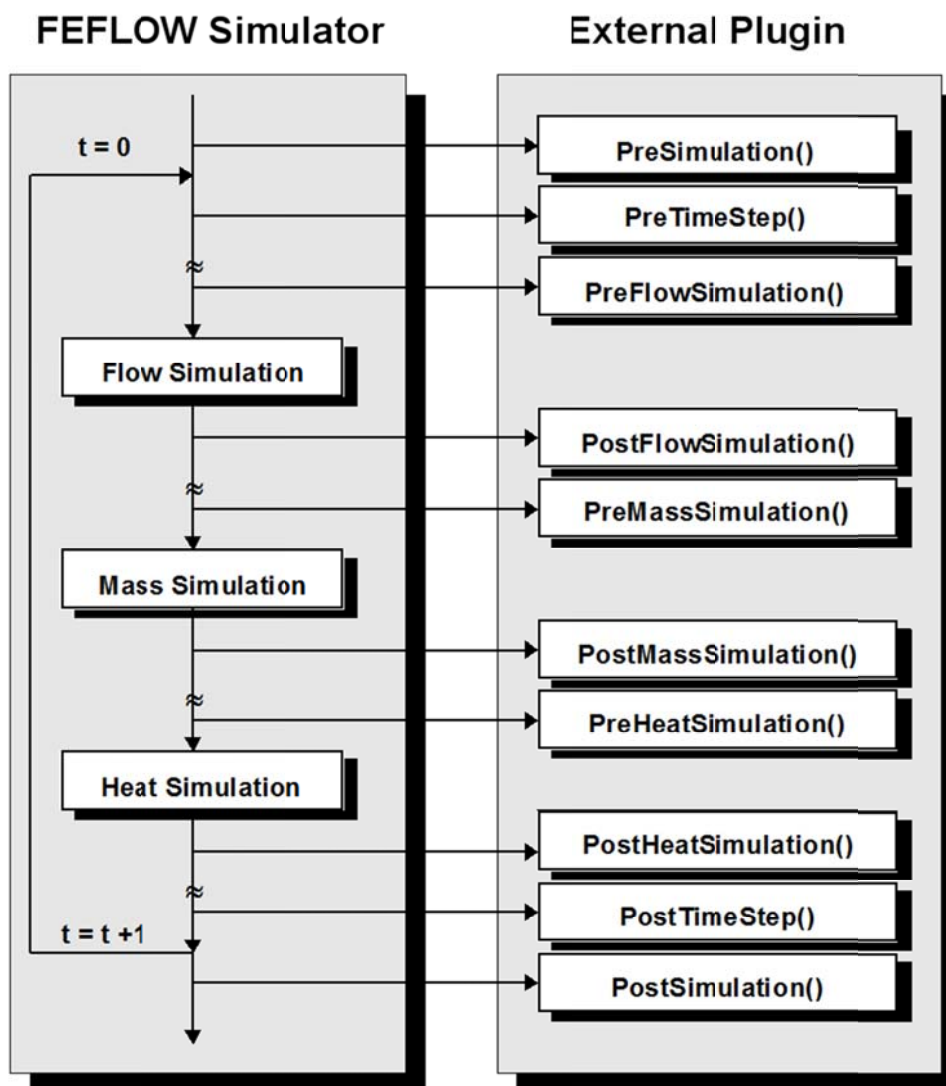
# MOCO Technical Solution

## Introduction

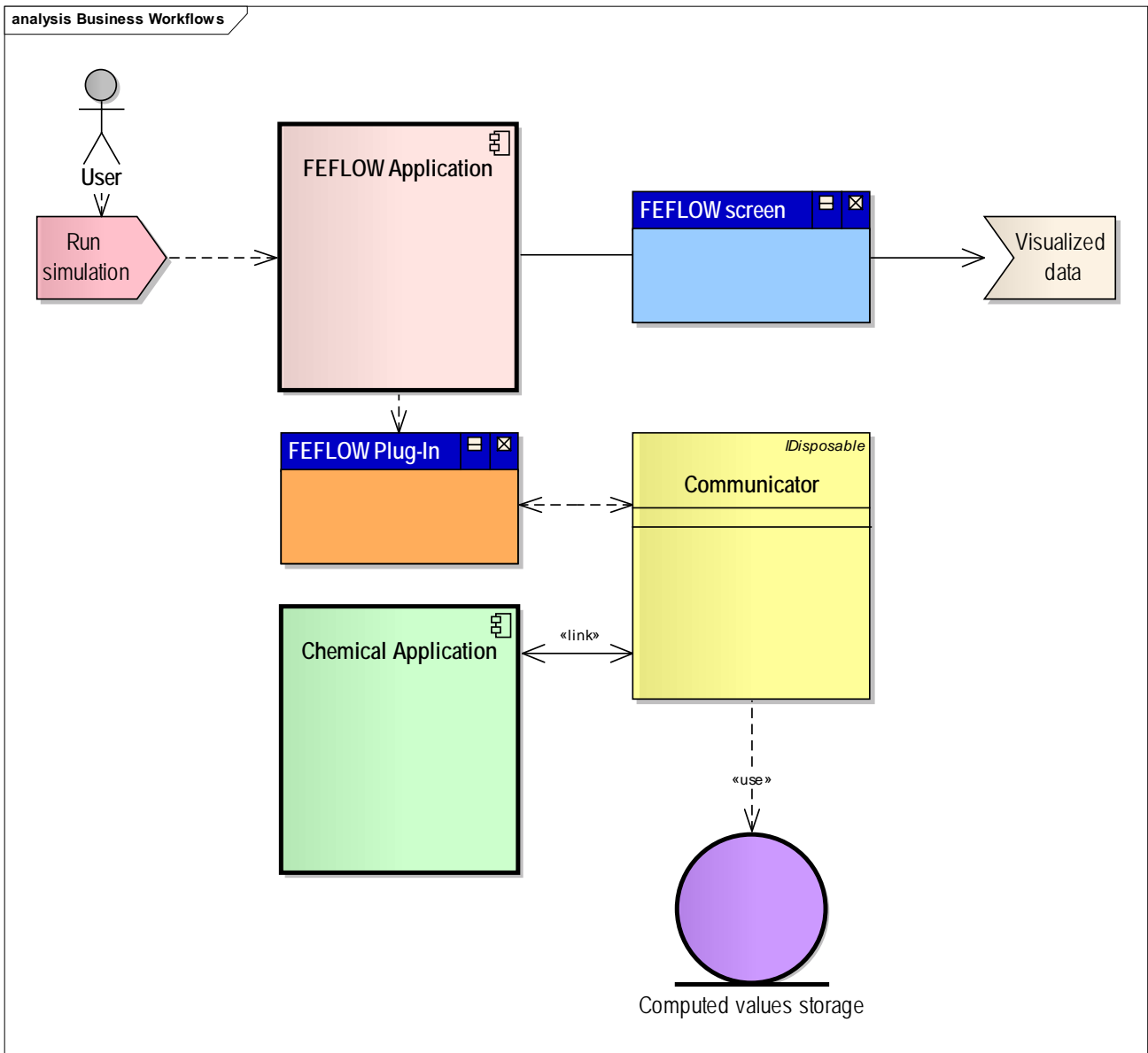
This technical solution solves the communication task between transport module (FEFLOW) and several reaction modules in common way (in our case only GWB).

## Solution

Communication of modules is handled through “Contaminants Mobility” FEFLOW Plug-in. Plug-in code is written on C++, but mixed-language programming is used inside. Plug-in PreSimulation() function is used for communicator initialization based on configuration files. Communication take place in PostTimeStep() function of plug-in.

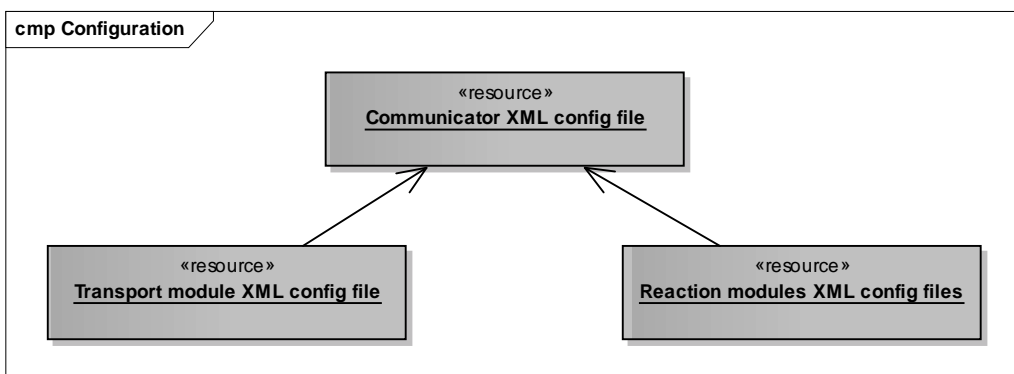


FEFLOW application is used also for results presentation in standard way. Computed value storage used for speedup return from chemical module. Computed values can be stored in Postgre database.



## Configuration

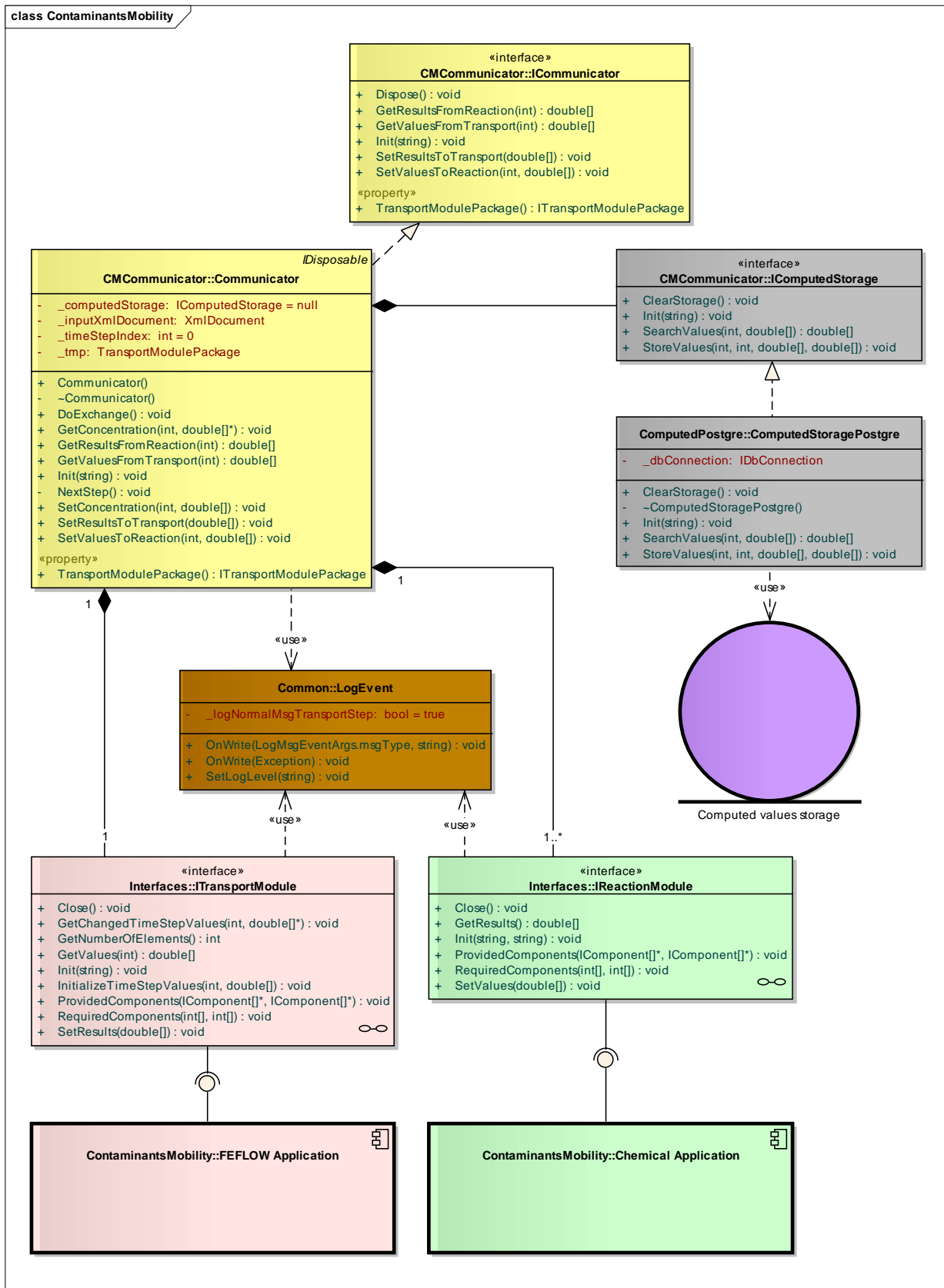
FEFLOW model is needed for simulation process. Corresponding configuration files should be prepared. Configuration data should relate to model chemical species and reaction module. Computed value storage defined in communicator configuration xml file if needed.



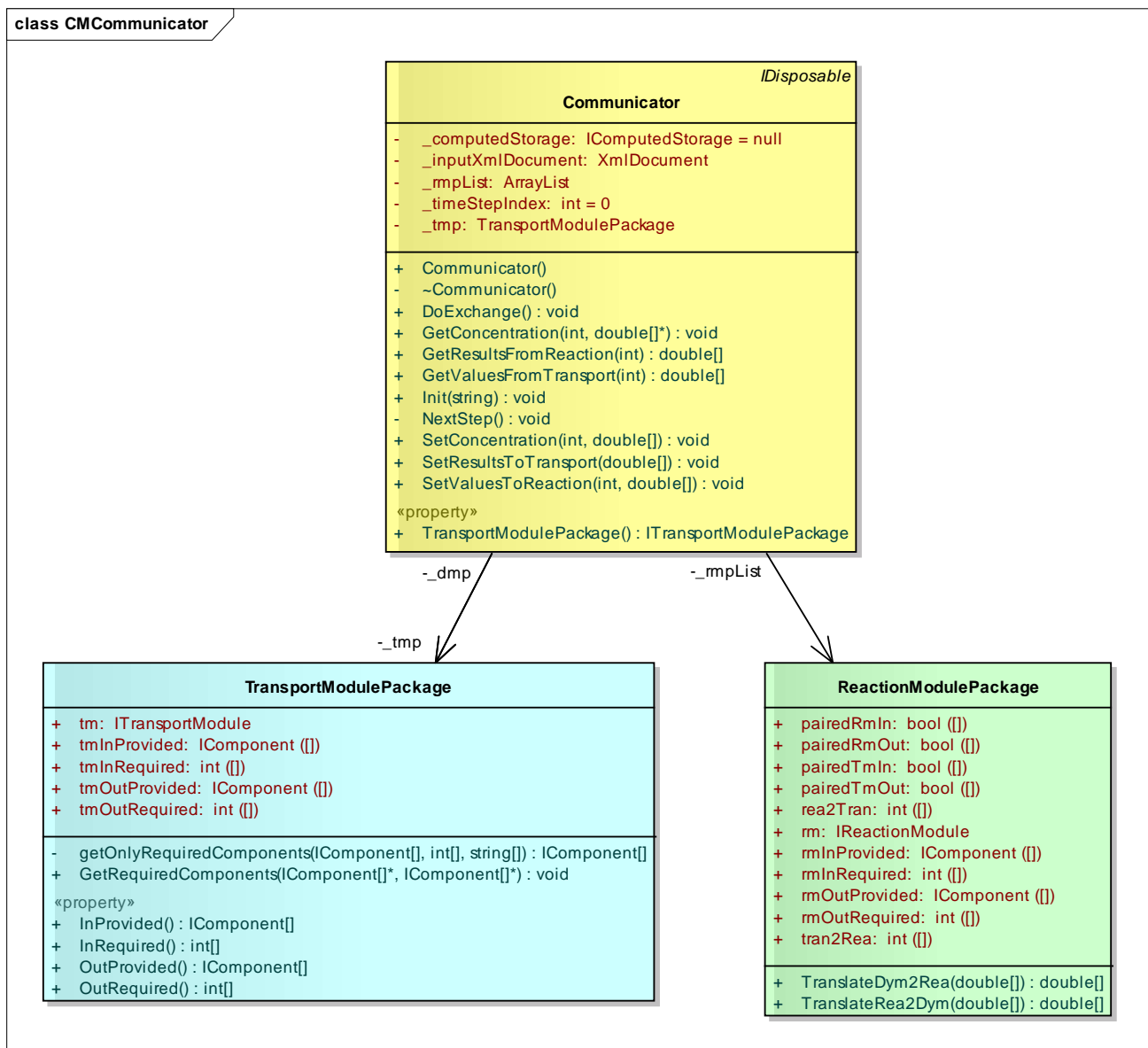


## Classes and interfaces

All behaviors of solution defined in interfaces. Classes implement correspond interfaces. This provides to developers a big flexibility in implementation.



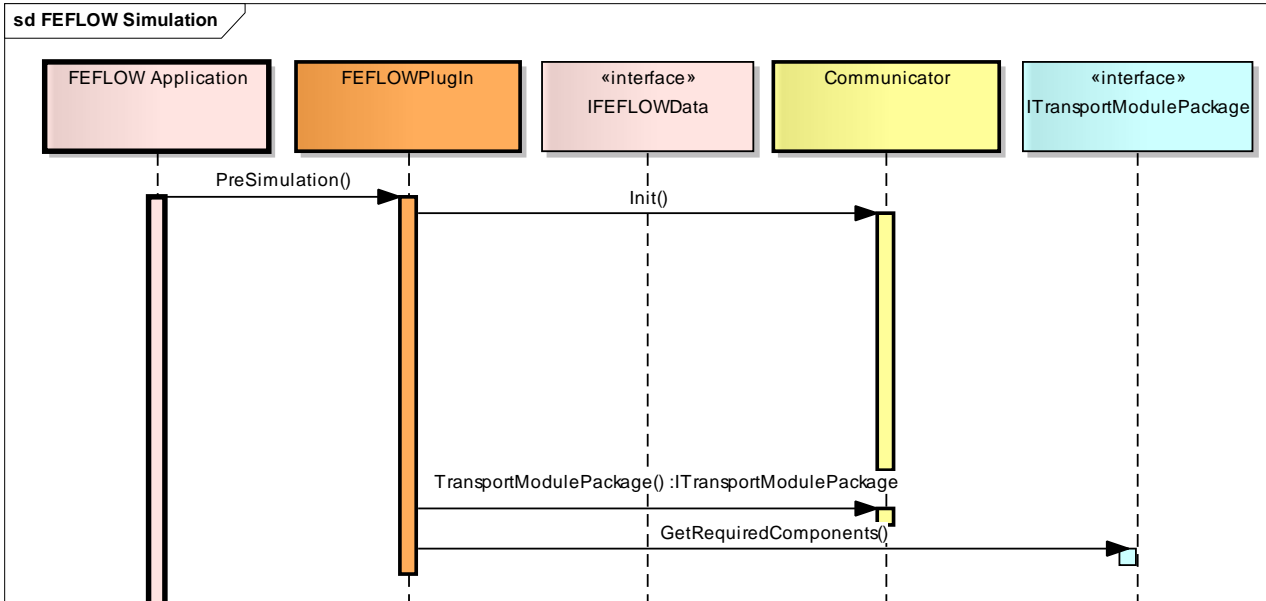
Main class Communicator contains transport and reactions modules. It holds definition of one transport and list of reaction modules.



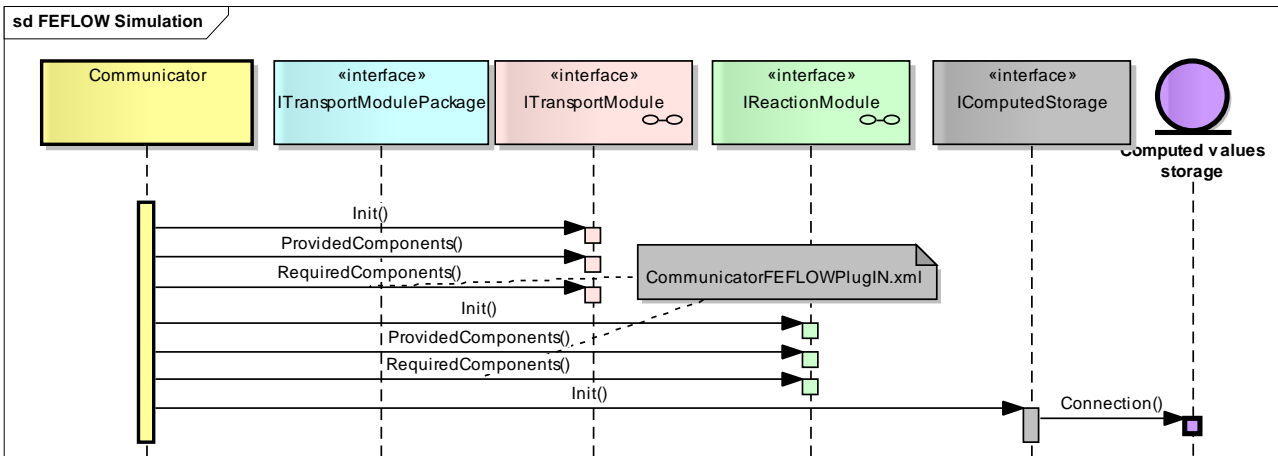
## Dynamic view, sequence diagram

Sequence diagram provide a view on interaction between a parts of solution.

Communicator initialization take place in PreSimulation() method of FEFLOW plug-in.

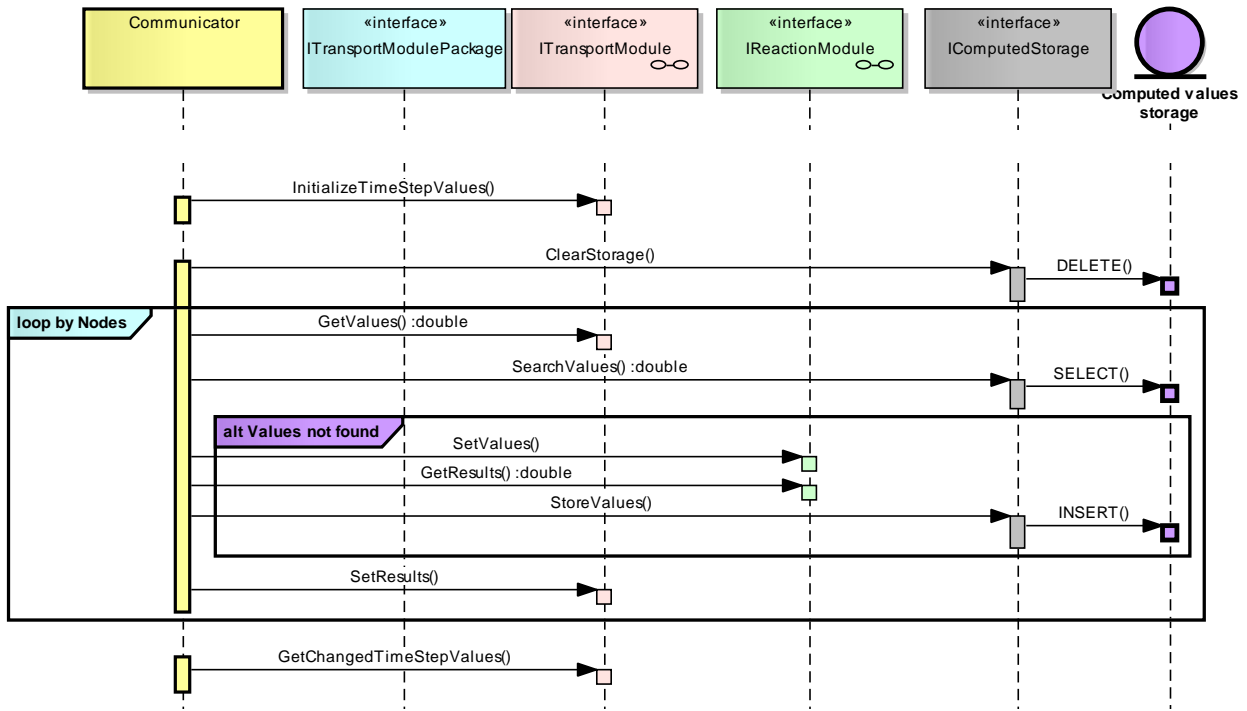


Communicator initializes transport and reaction modules based on xml configuration files. It also initializes computed storage with database connection if needed.



Time step loop controlled by FEFLOW plug-in via PostTimeStep() method. This method used by communicator for:

- Getting species concentration and boundary data such as temperature, time step from transport module
- Check if computed set of values exists for current concentrations and boundaries, if needed
- Put current concentrations and boundaries to reaction modules, if needed
- Store computed values, if needed
- Set back results from reaction modules to transport module

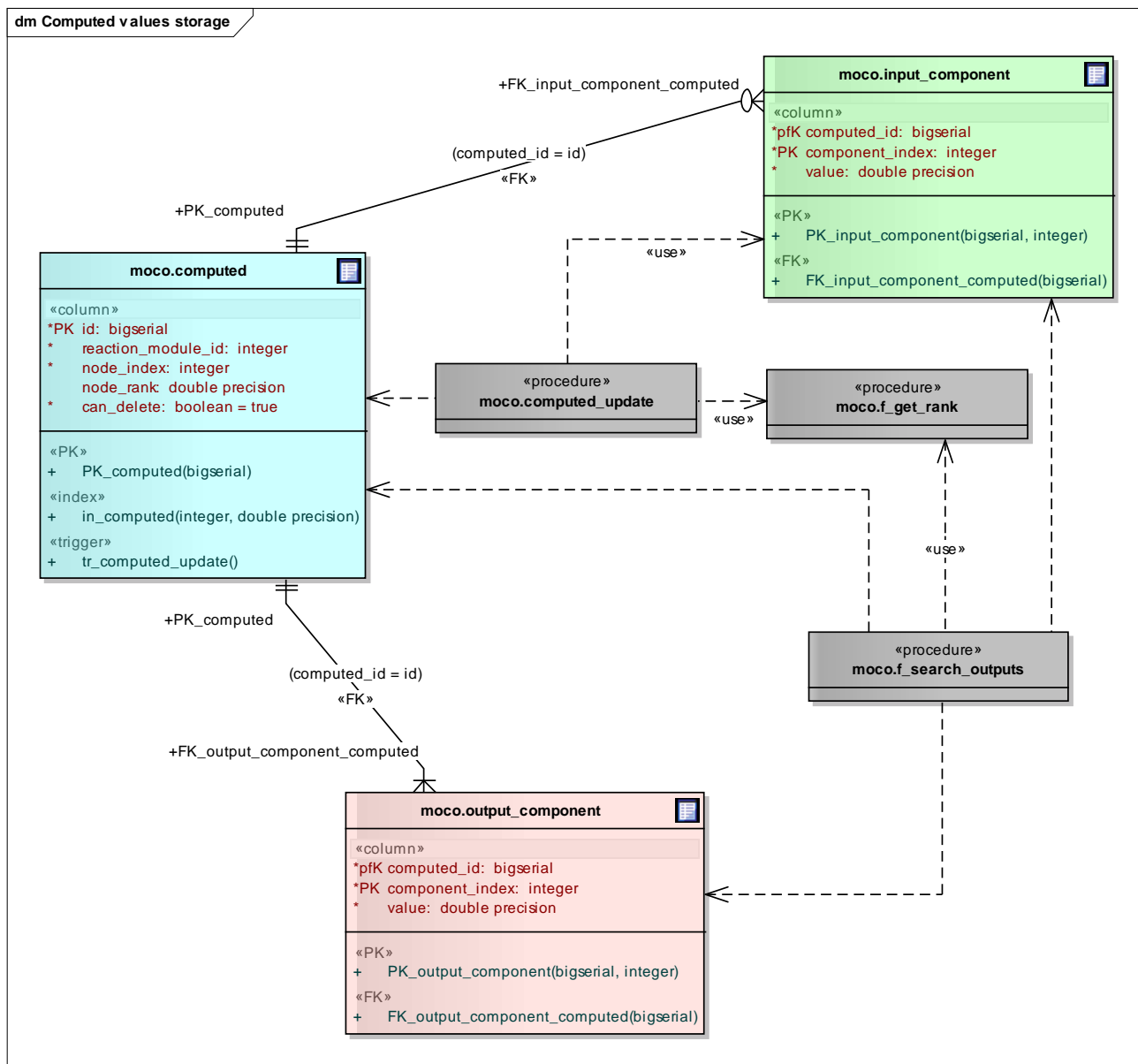


Complete sequence diagram:



## Computed value storage

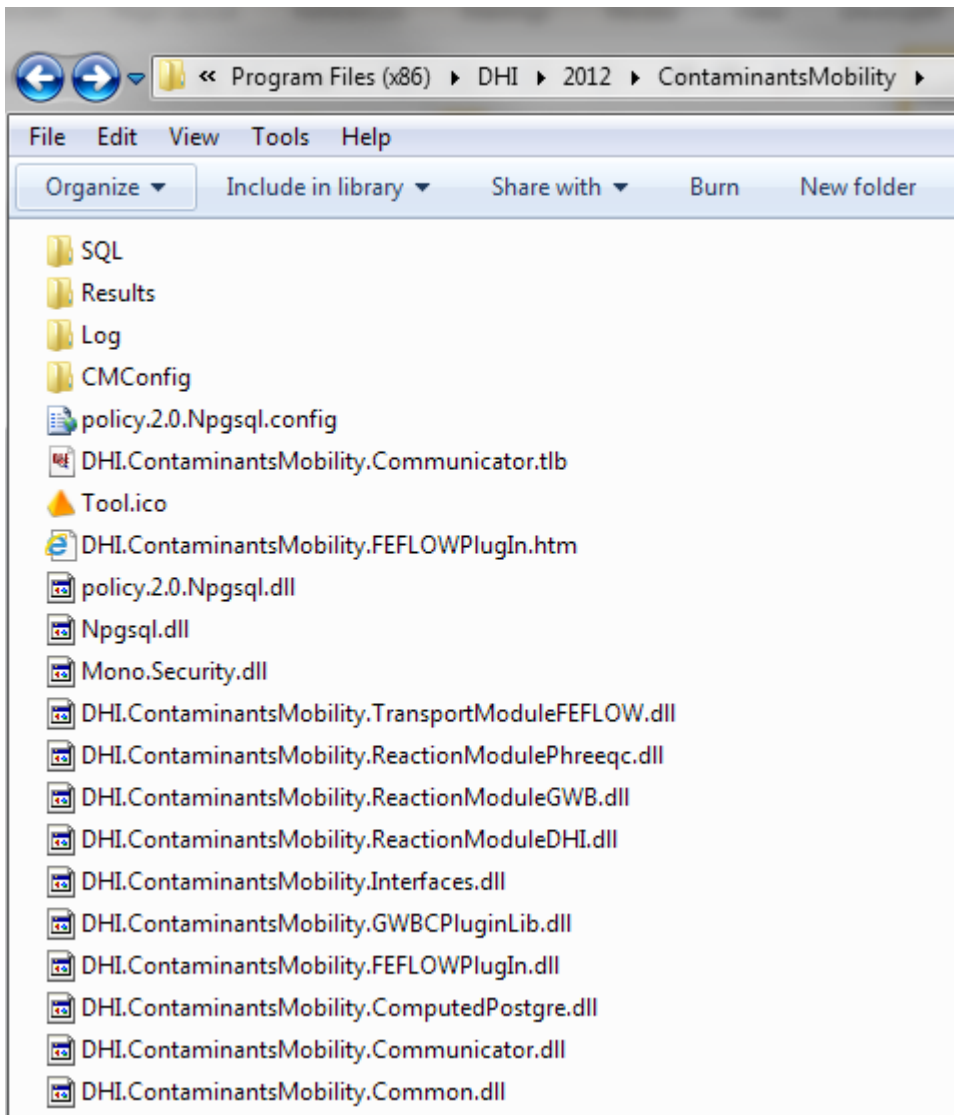
Database structure designed for storing computed values.



Moco.Input\_component table used for store input values which sets to reaction model. Output from reaction module is stored in moco.output\_component table. Few database procedures are created to make easier search correspond computed values. Procedure moco.f\_get\_rank calculates node rank based on node input values. Procedure moco.computed\_update updates node rank based value returned by moco.f\_get\_rank procedure. Procedure moco.f\_search\_outputs search for outputs based on inputs.

## Installation

Setup package provided for installation. Contaminant mobility folder contains binary files, configuration files in CMConfig folder after installation. Stored result and log files saved in corresponded folders after simulation run. DHI.ContaminantsMobility.FEFLOWPlugIn.dll is contaminants mobility FEFLOW plug-in.



## XML configuration files

Configuration files are located in CMConfig folder.

### [CommunicatorFEFLOWPlugIN.xml](#)

Detailed information can be found in Communicator.docx.

### [TransportModuleFEFLOWCaCO3.xml](#)

Detailed information can be found in TransportModuleFEFLOW.docx.

### [ReactionModuleGWBCaCO3.xml](#)

Detailed information can be found in ReactionModuleGWB.docx.

## SQL

Database installation requires in case of using computed storage. Set of sql command provided to initialize database structure for computed storage can be found in SQL folder (moco.SQL).

-- ..-----

```

-- Generated by Enterprise Architect Version 7.5.848
-- Created On : 09 10 2015
-- DBMS      : PostgreSQL
-- -----

-- Drop Tables, Stored Procedures and Views
DROP TABLE IF EXISTS moco.computed
;
DROP TABLE IF EXISTS moco.input_component
;
DROP TABLE IF EXISTS moco.output_component
;

-- Create Tables
CREATE TABLE moco.computed (
    id bigserial NOT NULL, -- Autoincremented index
    reaction_module_id integer NOT NULL, -- Reaction module identifier
    node_index integer NOT NULL, -- Index of node for which precomputed data is stored
    node_rank double precision, -- Rank of stored data for node
    can_delete boolean DEFAULT true NOT NULL
)
;
COMMENT ON COLUMN moco.computed.id
    IS 'Autoincremented index'
;
COMMENT ON COLUMN moco.computed.reaction_module_id
    IS 'Reaction module identifier'
;
COMMENT ON COLUMN moco.computed.node_index
    IS 'Index of node for which precomputed data is stored'
;
COMMENT ON COLUMN moco.computed.node_rank
    IS 'Rank of stored data for node'
;

CREATE TABLE moco.input_component (
    computed_id bigserial NOT NULL, -- Identificator of computed record
    component_index integer NOT NULL, -- Input component index
    value double precision NOT NULL -- Input component value
)
;
COMMENT ON COLUMN moco.input_component.computed_id
    IS 'Identificator of computed record'
;
COMMENT ON COLUMN moco.input_component.component_index
    IS 'Input component index'
;
COMMENT ON COLUMN moco.input_component.value
    IS 'Input component value'

```



```

;

CREATE TABLE moco.output_component (
    computed_id bigserial NOT NULL,
    component_index integer NOT NULL, -- Output component index
    value double precision NOT NULL -- Output component value
)
;
COMMENT ON COLUMN moco.output_component.component_index
    IS 'Output component index'
;
COMMENT ON COLUMN moco.output_component.value
    IS 'Output component value'
;

-- Create Primary Key Constraints
ALTER TABLE moco.computed ADD CONSTRAINT PK_computed
    PRIMARY KEY (id)
;

ALTER TABLE moco.input_component ADD CONSTRAINT PK_input_component
    PRIMARY KEY (computed_id, component_index)
;

ALTER TABLE moco.output_component ADD CONSTRAINT PK_output_component
    PRIMARY KEY (computed_id, component_index)
;

-- Create Indexes
CREATE INDEX in_computed
ON moco.computed (reaction_module_id, node_rank)
;

-- Create Foreign Key Constraints
ALTER TABLE moco.input_component ADD CONSTRAINT FK_input_component_computed
    FOREIGN KEY (computed_id) REFERENCES moco.computed (id)
ON DELETE CASCADE
;

ALTER TABLE moco.output_component ADD CONSTRAINT FK_output_component_computed
    FOREIGN KEY (computed_id) REFERENCES moco.computed (id)
ON DELETE CASCADE
;

-- Create Stored Procedures
CREATE OR REPLACE FUNCTION moco.computed_update()
    RETURNS trigger AS
$BODY$
    BEGIN

```

```

        select into NEW.node_rank moco.f_get_rank(NEW.reaction_module_id, VARIADIC array(select value from
moco.input_component where computed_id = NEW.id order by component_index));
    RETURN NEW;
END;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
ALTER FUNCTION moco.computed_update() OWNER TO postgres;
;

```

```

CREATE OR REPLACE FUNCTION moco.f_get_rank(IN module_id integer, VARIADIC arr double precision[])
RETURNS double precision AS
$BODY$
    SELECT sum($2[i]) FROM generate_subscripts($2, 1) g(i);
$BODY$
LANGUAGE sql VOLATILE
COST 100;
ALTER FUNCTION moco.f_get_rank(integer, double precision[]) OWNER TO postgres;
;

```

```

CREATE OR REPLACE FUNCTION moco.f_search_outputs(IN module_id integer, VARIADIC arr double precision[])
RETURNS SETOF moco.output_component AS
$BODY$
select moco.output_component.* from moco.output_component,
(
select moco.computed.id, abs(moco.computed.node_rank - moco.f_get_rank($1, VARIADIC ($2))) as diff from
moco.computed
where
moco.computed.reaction_module_id = $1
order by diff
limit 1
) as row
where computed_id = row.id and diff < 1e-9
order by component_index;
$BODY$
LANGUAGE sql VOLATILE
COST 100
ROWS 1000;
ALTER FUNCTION moco.f_search_outputs(integer, double precision[]) OWNER TO postgres;
;

```

```

-- Create Triggers
CREATE TRIGGER tr_computed_update
BEFORE UPDATE
ON moco.computed
FOR EACH ROW
EXECUTE PROCEDURE moco.computed_update(E'\x');
;

```