```
<Computer>CooLib.CooObject: Internet Browser
(COO.2249.100.1.130)</Computer>
<Login>CooLib.CooObject: Administrator, System
(COO.2249.100.1.14)</Login>
<ModulName>WSOperations</ModulName>
<MethodName>WEBServiceOperations</MethodName>
<SourceData><?xml version="1.0" encoding="UTF-8"?><
    executeOperationsReqest
    xmlns="urn:schemas.fabasoft.com:webswc:wsoperations"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><guid
    xmlns="">bd99e05b-1951-445b-a889-0c4933a98951</guid><appid
    xmlns="">WSTest</appid><userid xmlns="">MyName</userid><time
    xmlns="">2014-04-23T16:26:30.4304794+02:00</time><transaction
    xmlns=""><operation
    xmlns:q1="urn:schemas.fabasoft.com:webswc:wsoperations"
    xsi:type="q1:ObjectType" guid="ea0b5345-e33a-4ae9-baa7-
    a09a6330e892" operation="create" fileName=""
    objClass="SKFSCGOV@103.510:SubjectAreaFileSK"><property
    reference="COOELAK@1.1001:filesubj"><value xsi:type="q1:StringType"
    index="0" value="Moja
    vec"/></property></operation></transaction></executeOperationsReqes
    t> </SourceData>
<GUID>bd99e05b-1951-445b-a889-0c4933a98951</GUID>
<SourceSystem>WSTest</SourceSystem>
<ExternalUser>MyName</ExternalUser>
<CallTime>2014-04-23T16:26:30.4304794+02:0</CallTime>
<Progress>16:26:30.319 -> Action: Create16:26:30.358 -> Commit
    start16:26:33.182 -> Commit end - Obj count =1</Progress>
<ResultCode>0</ResultCode>
<ResultDescription>Ok</ResultDescription>
<ResultData><?xml version="1.0"?><mstns:resultOperationsResponse
    xmlns:jaxb="http://java.sun.com/xml/ns/jaxb"
    xmlns:mstns="urn:schemas.fabasoft.com:webswc:wsoperations"
    xmlns:xs="http://www.w3.org/2001/XMLSchema-instance">
    <guid>bd99e05b-1951-445b-a889-0c4933a98951</guid>
    <appid>UWS</appid> <time>2014-04-23T16:26:33.191</time>
    <computerid>COO.2249.100.1.130</computerid>
    <operationsCount>1</operationsCount> <transaction
    xs:type="mstns:ResultTransactionType"> <operation
    xs:type="mstns:ObjectType" guid="ea0b5345-e33a-4ae9-baa7-
    a09a6330e892" cooaddr="COO.2249.100.2.1110545"
    objName="00005/2014/Administration"
    objClass="SKFSCGOV@103.510:SubjectAreaFileSK"/> </transaction>
    </mstns:resultOperationsResponse> </ResultData>
</Log>
```

2.5 Dostupnosť UWS

Webové služby sú dostupné prostredníctvom protokolu HTTP s rozšírením WEB DAV. Systém Fabasoft prezentuje WEB DAV na adrese HTTP://<webserver>/<vdir>/fscdav/

Získanie definície webových služieb (WSDL) je možné na adrese

```
http://<webserver>/<vdir>/fscdav/wsdl?WEBSVC=COO.103.510.1.751196
```

alebo

```
http://<webserver>/<vdir>/fscdav/wsdl?WEBSVC=SKWEBSVC_103_510_UniversalWebService
```

kde `webserver` je názov alebo IP adresa aj s portom webového servera a `vdir` je názov virtuálneho adresára IIS, spravidla má názov `fsc`.

Adresa pre volanie webovej služby je http://<webserver>/<vdir>/fscdav/wsdl?actions,

pričom v hlavičke volania musí byť zadefinovaný typ obsahu `Content-Type` ako `text/xml` a názov metódy volanej webovej služby `SOAPAction` pre univerzálnu webovú službu `http://schemas.fabasoft.com/object/SKWEBSVC@103.510:Execute`.

Príklad:

```
var xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");
xmlhttp.Open("POST", "http://websrv04.fsc/fscdav/wsdl?actions", false);
xmlhttp.setRequestHeader("Content-Type", "text/xml; charset=iso-8859-2");
xmlhttp.setRequestHeader("SOAPAction",
"http://schemas.fabasoft.com/object/SKWEBSVC@103.510:Execute");
xmlhttp.Send SOAPRequest.xml);
```

2.6 Dostupnosť obsahu prostredníctvom WEBDAV

Prístup k pracovnému adresáru prostredníctvom WEB DAV:

http://<webserver>/WEBDAV/.../subor.ext

Prístup k objektu s obsahom prostredníctvom WEB DAV v známej štruktúre objektov:

http://<webserver>/<vdir>/fscdav/webfolder/dalsia/struktura/objekt.ext

Prístup k hlavnému obsahu objektu s obsahom prostredníctvom WEB DAV so známou COO adresou:

http://<webserver>/<vdir>/fscdav/dav?OBJ=COOADRESA

Príklady

Funkcia pre načítanie obsahu z objektu – Java script:

```
function getContentFromCOO(cooaddr, serverlink)
{
   var request = new XMLHttpRequest();
   var fname = 'http://' + serverlink + '/fscdav/dav?OBJ=' + cooaddr;
   request.open("GET", fname, false);
   request.send();
   if(request.responseText == null)
     alert "Obsah objektu " + cooaddr + " je prazdny!" );
   return request.responseText;
}
```

Zápis obsahu do objektu

```
fname = 'http://' + server + '/fscdav/dav?OBJ=' + cooadresa;
obsah = 'Text obsah.';
response.open "PUT", fname, false);
response.send obsah.;
```

Príklad vloženia obsahu do objektu v C#

```
WebClient wc = new WebClient();

wc.Credentials = new System.Net.NetworkCredential ("login", "password", "domena");

            wc.UploadFile("http://" + server + "/fscdav/dav?OBJ=" + cooadresa, "PUT",
"c:\subor.txt");
```

2.7 Dostupnosť objektu podľa COOadresy

Externé systémy môžu priamo otvárať objekty vo Fabasofte, použitím URL, ktorá priamo zobrazí formulár objektu, podľa COO adresy v prostredí Fabasoftu.

http://server/fsc/mx/COO.xxxx.yyy.z.ccccccc

- server – IP adresa, alebo host name s portom alebo bez
- fsc - pool na loadbalancer, môže sa líšiť od prostredia
- mx – fixné
- coo adresa objektu, ktorý je potrebné zobraziť vo Fabasofte

Príklady:

http://testwebfsc.vuctt.sk/fsc/mx/COO.2091.100.3.128354

http://ediregtest/fsc/mx/COO.200.200.3.609597

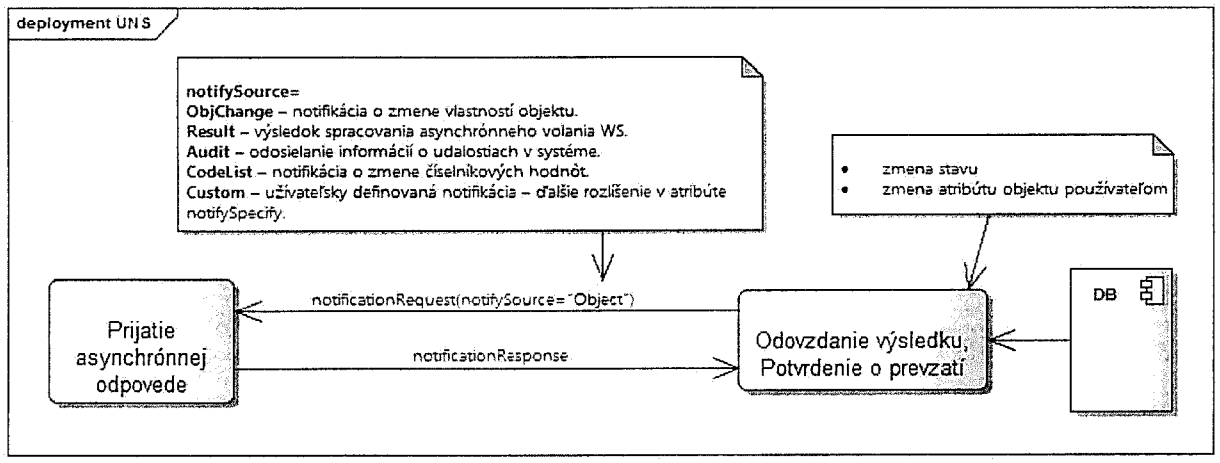
3 Univerzálna notifikačná služba (UNS)

UNS je komunikačný kanál, ktorým Fabasoft zasiela informácie o vzniknutých udalostiach do externých systémov.

3.1 Architektúra UNS

Aby bolo možné využívať UNS, je potrebné, aby bolo na strane externého (notifikovaného) systému implementované (vystavené) rozhranie pre príjem vstupného notifikačného volania.

V rámci UNS sa používa synchrónne spracovanie volaní.



4 Objektový model

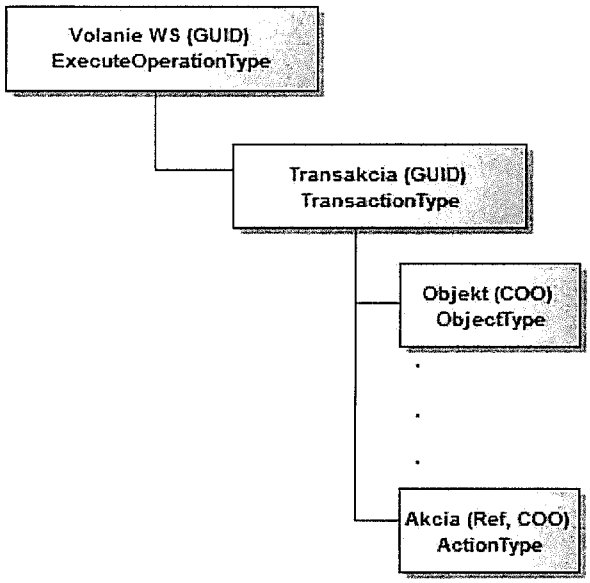
4.1 Elementy pre univerzálnu webovú službu

UWS používa **pre všetky spôsoby volania** (synchrónne, asynchrónne aj jednosmerné) **rovnaký vstupný a výstupný element**:

```
resultOperationsResponse response = UWS(executeOperationsReqest request)
```

4.1.1 Vstupný element – executeOperationsReqest

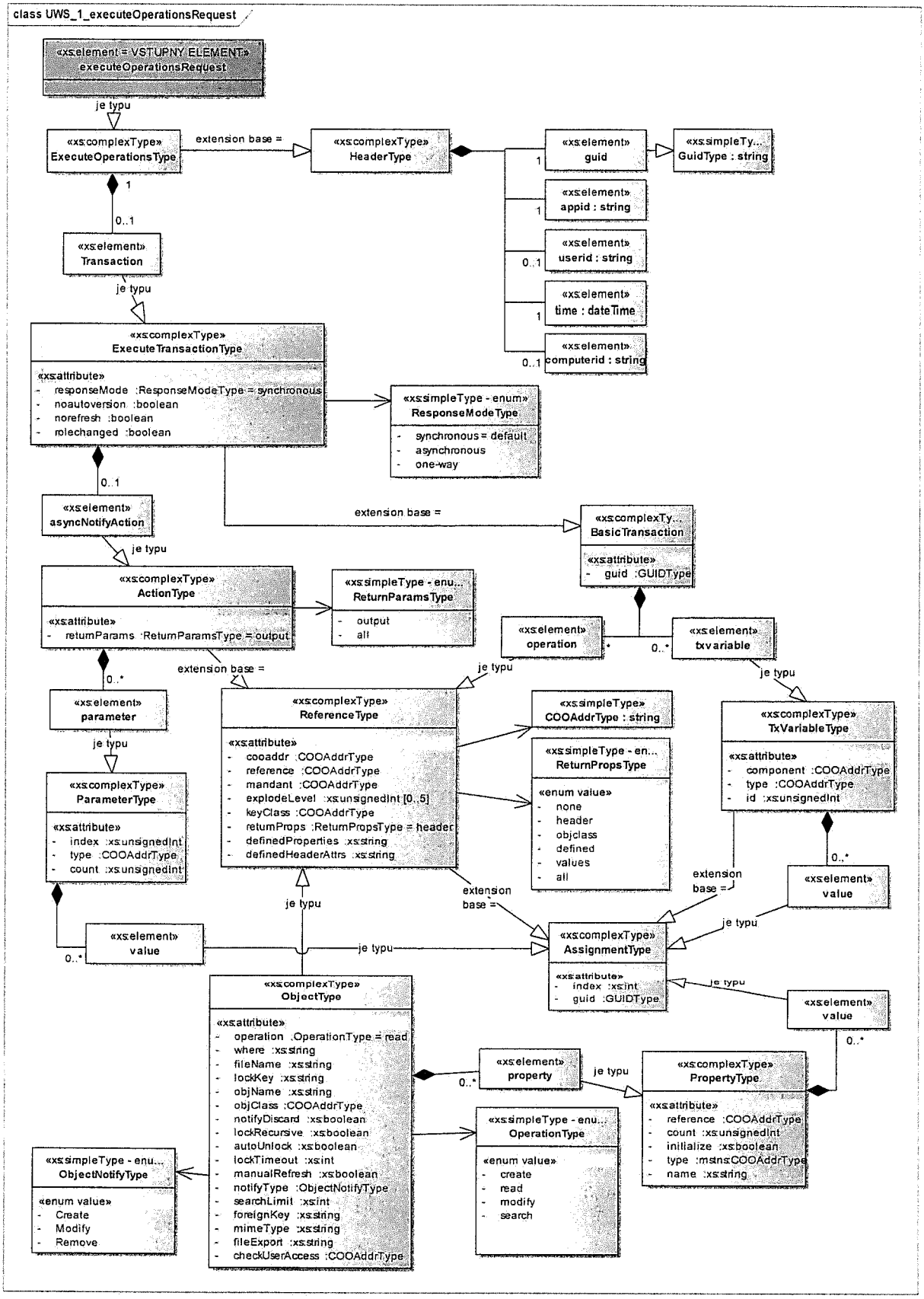
Spôsob spracovania, sekvencia vykonávaných operácií a množstvo a štruktúra výsledných údajov je daná štruktúrou zapísaných dát v tomto elemente. Vstupný element obsahuje údaje určené hlavne k logovaniu:



- `guid` – jedinečný identifikátor daného volania, povinný element. Slúži k identifikovaniu volania v logovacom systéme a návratovej správe (potvrdení prevzatia alebo synchrónnej odpovedi).
- `appid` – skratka aplikácie, ktorá volá UWS, povinný element. Používa sa len v logovacom systéme, uľahčuje to vyhľadávanie v logoch.
- `userid` – identifikácia transakčného používateľa, ktorý vyvolal volanie UWS na strane externého systému, nepovinný element. Najčastejšie sa ako identifikátor používa login používateľa. Tento atribút sa zapisuje do logovacieho systému pre lepšie filtrovanie, hlavne v systémoch, ktoré volajú UWS so serverovými účtami (technický používateľ).
- `time` – čas volania na klientskej strane volania, povinný element. Používa sa pre zápis do logov a na výpočet dĺžky trvania prenosu. Tento čas by mal byť čo najbližšie odoslaniu požiadavky (volaniu služby).
- `computerId` – nepovinný element, identifikácia počítača alebo servera, z ktorého sa realizuje volanie. Použije sa pri zápise do logov a pre monitorovanie výkonnosti systému.

Najdôležitejším prvkom je element `transaction`, ktorý je typu `ExecuteTransactionType`. Slúži pre zápis samotnej transakcie na spracovanie, je to nepovinný element. Volanie UWS bez naplnenia transakcie nemá zmysel, ale je možné ho použiť na testovanie funkčnosti UWS napr. pre load balancer.

Nasledujúci obrázok obsahuje model tried popisujúci elementy, atribúty a dátové typy v rámci vstupného elementu `executeOperationsReqest`.



Vysvetlivky: oranžová farba = základný element vstupného XML súboru; žltá farba = ostatné elementy vstupného XML súboru; zelená farba = typy definované vo WSDL; modrá farba = enumerácie;

4.1.1.1 Transakcia – ExecuteTransactionType

Element transakcie obsahuje údaje pre nastavenie transakcie pred jej spustením a zoznam samotných operácií na spracovanie:

- `guid` – identifikátor transakcie, na identifikáciu počas celého cyklu transakcie.
  - **Synchrónne volanie** – `guid` sa vracia v odpovedi v zodpovedajúcom elemente transakcie.
  - **Asynchrónne volanie** – `guid` sa vracia v odpovedi potvrdzujúcej prevzatie požiadavky v zodpovedajúcom elemente a tiež v transakcii na prevzatie výsledku po vykonaní. Ak sa znovu vyskytne volanie s rovnakým identifikátorom transakcie, aké je už uložené v DB pre spracovanie, bude iba potvrdené prevzatie bez založenia novej transakcie na spracovanie a v potvrdení prevzatia bude (v atribúte `progress` v rámci elementu `ackTransaction`) nastavený aktuálny status úlohy. Pri volaní externého systému na prevzatie výsledkov spracovania bude rovnako tento identifikátor použitý v zodpovedajúcom elemente.
  - **Jednosmerné volanie** – pre `guid` platia rovnaké pravidlá ako pre asynchrónne volanie.
- `responseMode` – spôsob spracovania transakcie:
  - `synchronous` – synchrónna transakcia, výsledok spracovania je k dispozícii vo výstupnom XML (`ResultTransactionType`).
  - `asynchronous` – asynchrónna transakcia, systém Fabasoft prevezme v transakcii zoznam operácií na spracovanie a v synchrónnej odpovedi (`AckTransactionType`) potvrdí prevzatie. Následne prebehne spracovanie, ktorého začiatok môže závisieť od vyťaženia systému a množstva predchádzajúcich požiadaviek. Po spracovaní je volaná predpísaná akcia (definovaná vo vstupnom volaní v elemente `asyncNotifyAction`).
  - `one-way` – jednosmerná asynchrónna transakcia. Má rovnaký postup spracovania ako asynchrónna transakcia s tým rozdielom, že po spracovaní sa nevracia výsledok spracovania. Postup spracovania však možno sledovať rovnako ako pri asynchrónnej transakcii.
- `noautoversion` – nepovinný atribút typu `boolean`, pri hodnote `true` zabezpečuje zákaz vytvárania automatických verzií spracovávaných objektov.
- `norefresh` – nepovinný atribút typu `boolean`, pri hodnote `true` zabezpečuje zákaz refrešovania objektov po commite transakcie.
- `rolechanged` – nepovinný atribút typu `boolean`.
- `asyncNotifyAction` – element je typu `ActionType` a pre asynchrónnu transakciu musí obsahovať akciu pre odovzdanie výsledkov, pre ostatné je nepovinný.

Okrem nastavenia atribútov transakcie je možné pred vykonaním nastaviť aj transakčné premenné do elementu `txvariable`.

Zoznam operácií zasielaných na spracovanie sa zapisuje do elementu `operation`. Operácie nad objektmi a volania akcií sa zapisujú v poradí, v akom sa majú vykonať a ich vykonávanie je sekvenčné.

Všetky operácie sa v systéme Fabasoft vykonajú v rámci jednej transakcie a v prípade zlyhania alebo iného nekorektného ukončenia niektorej operácie je celá transakcia zrušená (rollback) a všetky zúčastnené objekty a ich vlastnosti sú vrátené do pôvodného stavu. Niektoré operácie však môžu byť realizované mimo samotnej transakcie, tieto zostanú aj po

zrušení (roll-back) transakcie v zmenenom stave (napr. práca so súborovým systémom). Preto pri zložitejších operáciách je nutná analytická konzultácia, prípadne špecifikácia kompenzácií.

Do elementu `operation` je možné zapisovať objekty typu `ObjectType` pre operácie nad objektom alebo `ActionType` pre volanie akcie.

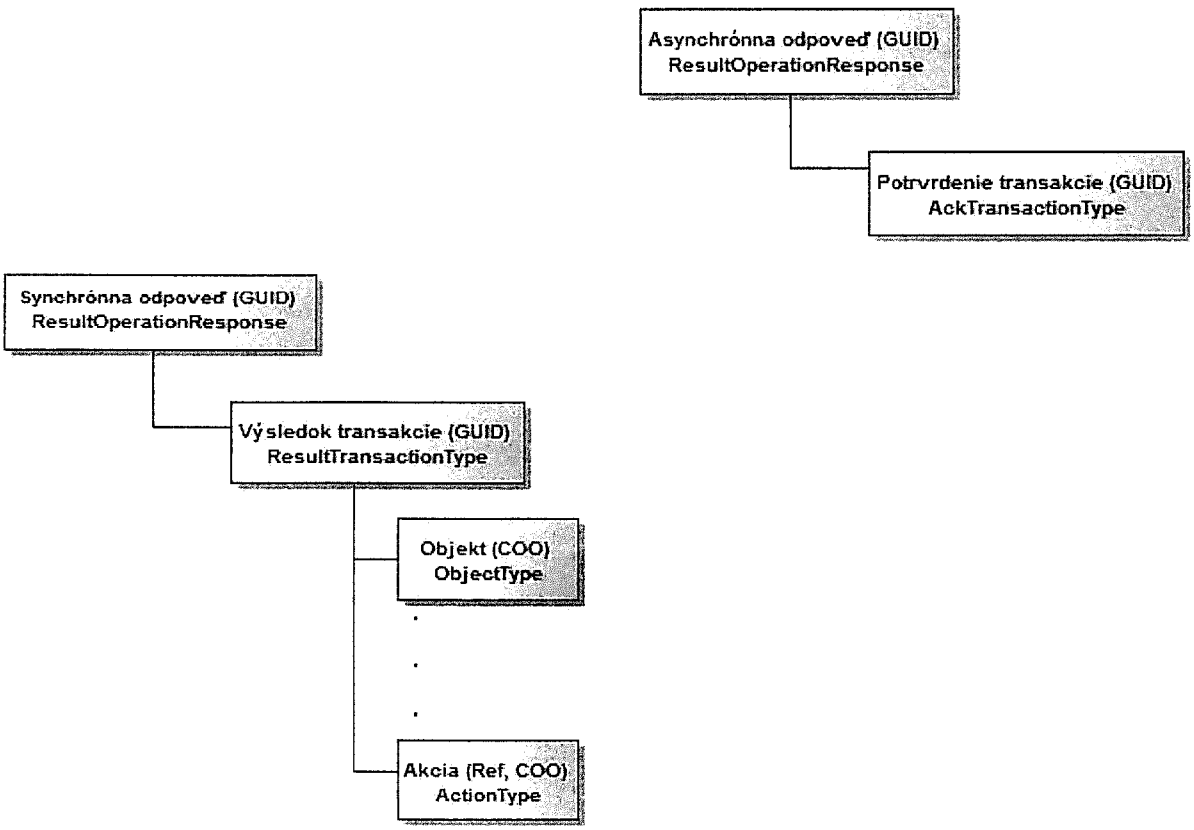
4.1.2 Výstupný element – resultOperationsResponse

Výstupný element `resultOperationsResponse` definuje štruktúru odpovede na vstupné volanie `executeOperationsRequest`.

Výstupný element rovnako ako vstupný obsahuje údaje určené hlavne na logovanie:

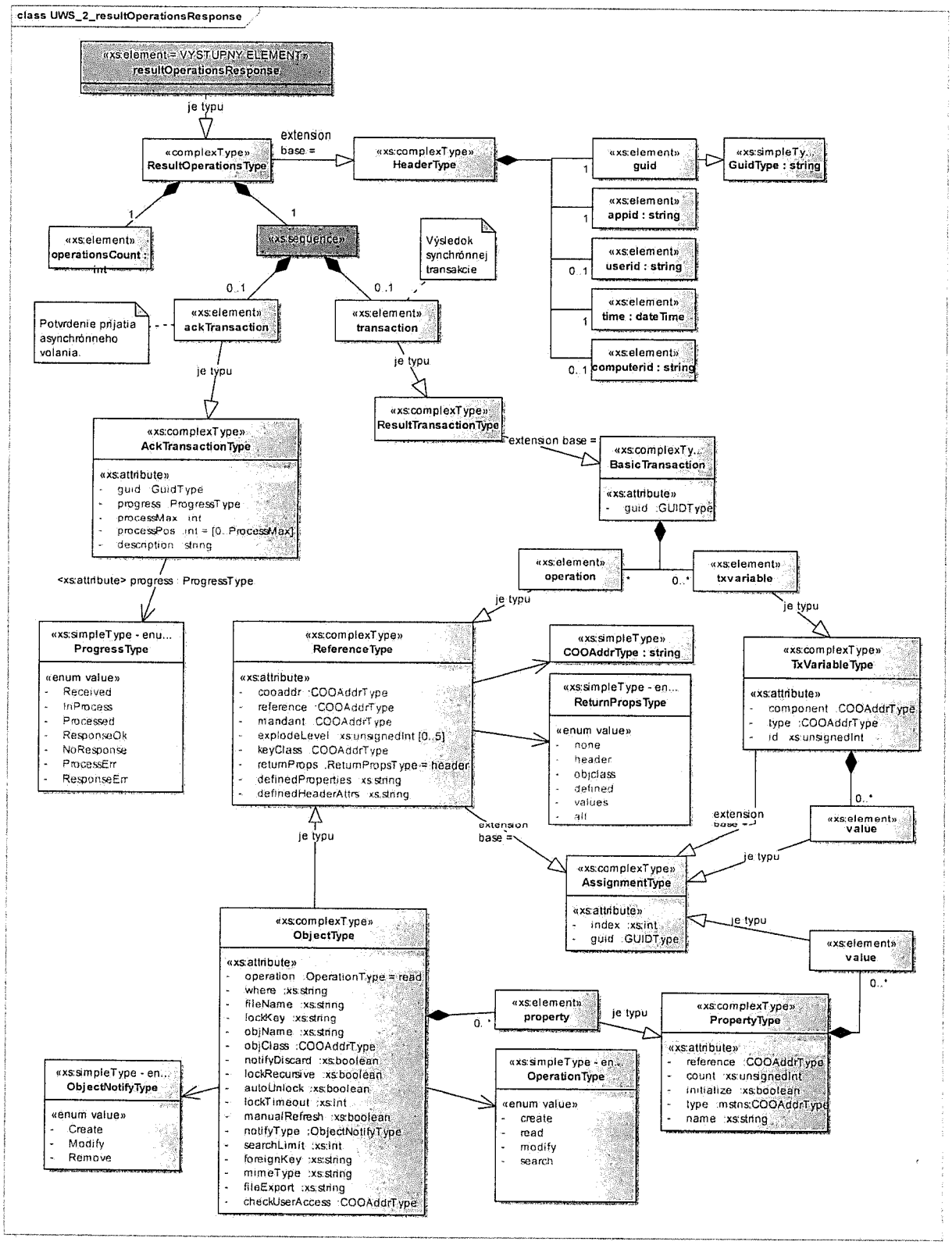
- `guid` – jedinečný identifikátor volania, povinný element, vracia sa hodnota identifikátora prevzatá zo vstupného elementu.
- `appid` – skratka aplikácie, povinný element, v odpovedi bude obsahovať reťazec „UWS".
- `computerid` – nepovinný element, identifikácia webového servera, na ktorom prebehlo spracovanie alebo ktorý prevzal transakciu na spracovanie.
- `time` – čas pred odoslaním odpovede. Používa sa pre zápis do logov a na výpočet dĺžky trvania prenosu.
- `operationsCount` – počet výsledkov operácií vrátených po spracovaní transakcie. Povinný informačný element uvádzaný len pre zrýchlenie spracovania.

V prípade, že ide o odpoveď na synchrónnu transakciu, bude obsahovať element `transaction`, typu `ResultTransactionType` ktorý v sebe nesie výsledok spracovanej transakcie, v ostatných prípadoch je prázdny.



Pri volaní asynchrónnej alebo jednosmernej transakcie bude obsahovať element `ackTransaction`, typu `AckTransactionType`, ktorý obsahuje potvrdenie a stav spracovania asynchrónneho a jednosmerného volania, v ostatných prípadoch je prázdny.

Nasledujúci obrázok obsahuje model tried popisujúci elementy, atribúty a dátové typy v rámci výstupného elementu `resultOperationsResponse`.



4.1.2.1 Výsledok vykonania transakcie – ResultTransactionType

Vrátená transakcia obsahuje identifikáciu v atribúte `guid`. V prípadoch keď sa vracia výsledok rovnakej transakcie tento atribút má rovnakú hodnotu. Pri synchrónnom volaní systém nekontroluje duplicitu identifikátorov transakcií.

Ak boli v transakcii vrátené transakčné premenné, budú naplnené v elemente `txvariable`. Štruktúra je rovnaká ako pri transakčných premenných vo vstupnej transakcii.

Zoznam vrátených objektov a akcií je zapísaný v elemente `operation`. Poradie objektov a akcií v tomto elemente nemusí zodpovedať poradiu, ako boli jednotlivé operácie vykonávané. Pre identifikáciu zodpovedajúcich objektov je potrebné použiť atribút `guid`. Popis atribútov a elementov objektov je v kapitole Operácie nad objektom. Popis atribútov a elementov akcie je v kapitole Operácia volanie akcie.

4.1.2.2 Potvrdenie prevzatia transakcie – AckTransactionType

UWS zasiela potvrdenie o prevzatí v prípade, že vstupné XML obsahovalo transakciu na asynchrónne spracovanie (asynchrónna alebo jednosmerná transakcia). Potvrdenie obsahuje informáciu o stave spracovávanej transakcie ako aj progres samotného vykonávania:

- `progress` – enumerácia, vyjadruje stav spracovávania transakcie:
  - `Received` – nová transakcia, čaká na začatie spracovania.
  - `InProcess` – transakcia je spracovávaná. Pri spracovaní môže byť postup prenášaný cez atribúty `ProcessPos`, `ProcessMax` a `Description`.
  - `Processed` – transakcia bola vykonaná, výsledok nebol odovzdaný.
  - `ResponseOk` – spracovanie transakcie ukončené, výsledok transakcie bol odovzdaný.
  - `NoResponse` – spracovanie transakcie s jednosmerným volaním ukončené.
  - `ProcessErr` – spracovanie transakcie skončilo s chybou.
  - `ResponseErr` – odovzdávanie výsledkov skončilo s chybou.
- `processPos` – aktuálny krok v procese.
- `ProcessMax` – maximálny počet krokov v procese.
- `Description` – popis alebo výstup k aktuálnemu kroku.

Pre opakované zistenie stavu spracovania asynchrónnej transakcie je možné zavolať `executeOperationsReqest`, pričom samotný element `transaction` bude obsahovať prázdnu transakciu len s vyplneným atribútom `guid`. (T.j. `guid` element v hlavičke volania bude obsahovať unikátny identifikátor daného volania, ale `guid` atribút v rámci elementu `transaction` bude obsahovať pôvodné `guid` danej transakcie, s ktorým bola do UWS odoslaná.)

Pri stave spracovania `progress = InProcess` je možné zobrazovať postup spracovania v grafickej forme (atribúty `ProcessPos`, `ProcessMax`, `Description`).

4.2 Elementy pre univerzálnu notifikačnú službu

UNS používa pre notifikáciu jeden vstupný element a jeden výstupný element. Pre prijatie univerzálnej notifikácie musí mať notifikovaný systém implementovanú (vystavenú) webovú službu s príslušným vstupným a výstupným elementom:

```
response = UNS(request)
```

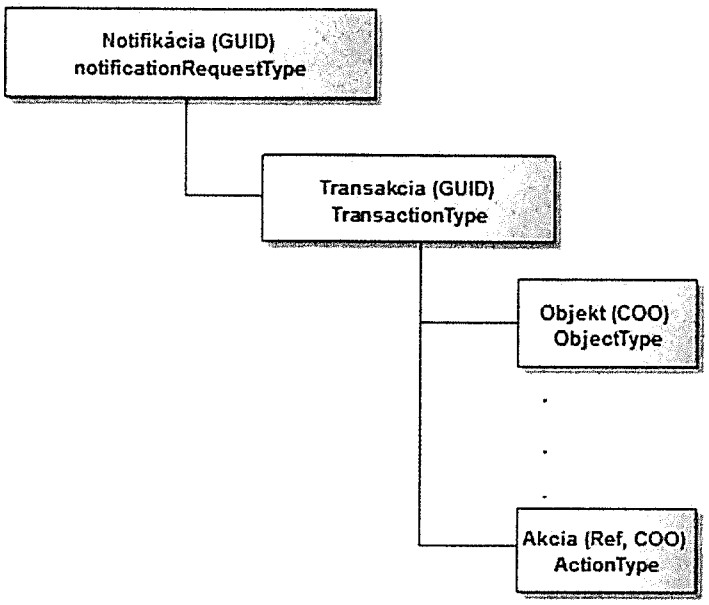
4.2.1 Vstupný element notifikácie – notificationRequest

Pre štruktúru vstupného notifikačného elementu `notificationRequest`, ktorý je typu `notificationRequestType`, platia analogické pravidlá ako pre výstupný element pri synchrónnom spracovaní (`resultOperationsResponse`), to znamená:

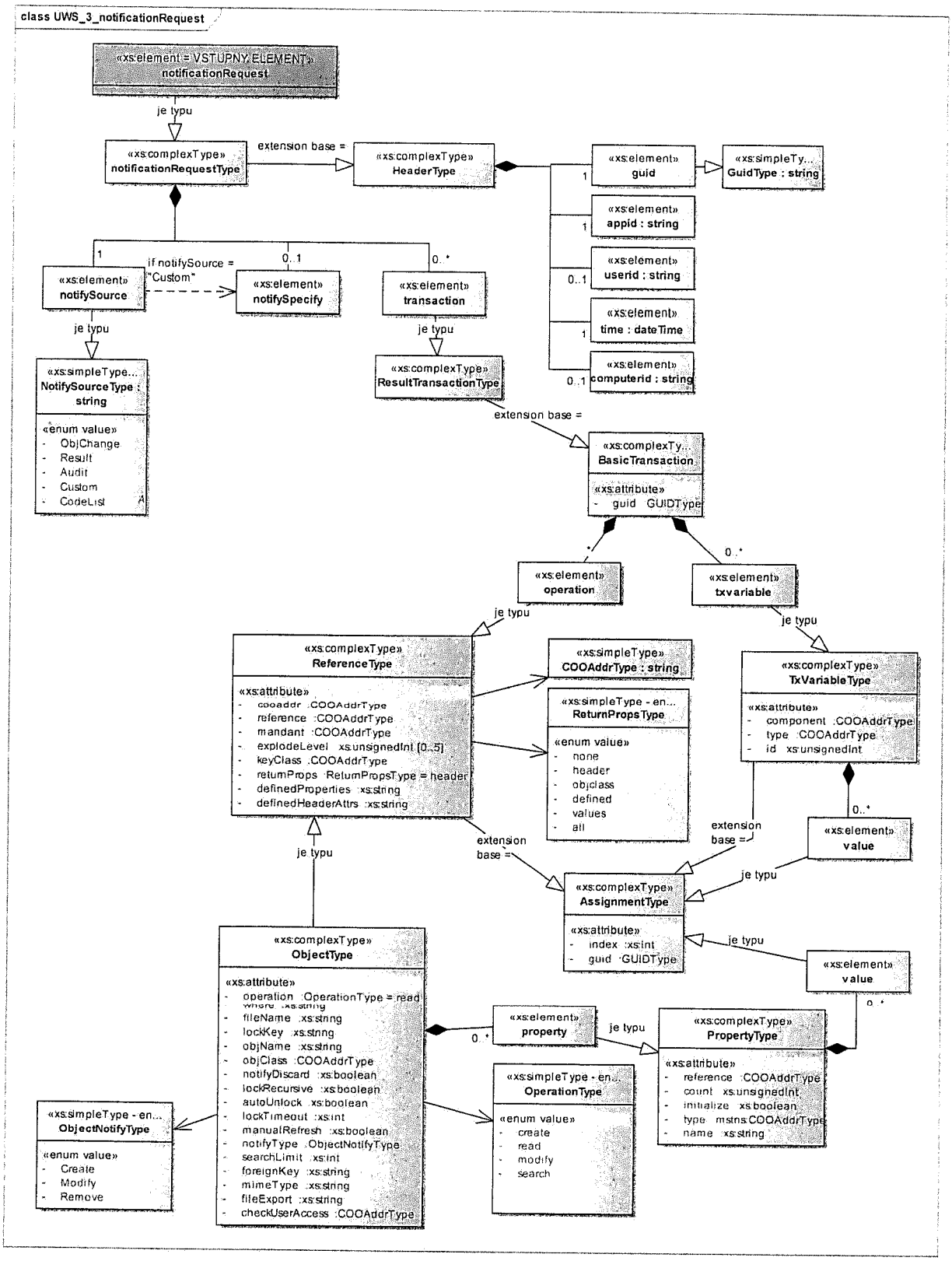
- hlavička vstupného volania je definovaná typom `HeaderType` a obsahuje elementy `guid`, `appid`, `userid`, `time`, `computerid`;
- transakcia (element `transaction`) vstupného notifikačného elementu je typu `ResultTransactionType`;

Typ zasielaných informácií je definovaný elementom `notifySource` (typu `NotifySourceType`), ktorý špecifikuje typ notifikácie nasledovne:

- `ObjChange` – notifikácia o zmene vlastností objektu.
- `Result` – výsledok spracovania asynchrónneho volania WS (výsledok spracovania executeOperationsRequest).
- `Audit` – odosielanie informácií o udalostiach v systéme.
- `CodeList` – notifikácia o zmene číselníkových hodnôt.
- `Custom` – užívateľsky definovaná notifikácia. Typ užívateľskej notifikácie je možné definovať v elemente `notifySpecify`.



Nasledujúci obrázok obsahuje model tried popisujúci elementy, atribúty a dátové typy v rámci výstupného elementu `resultOperationsResponse`.

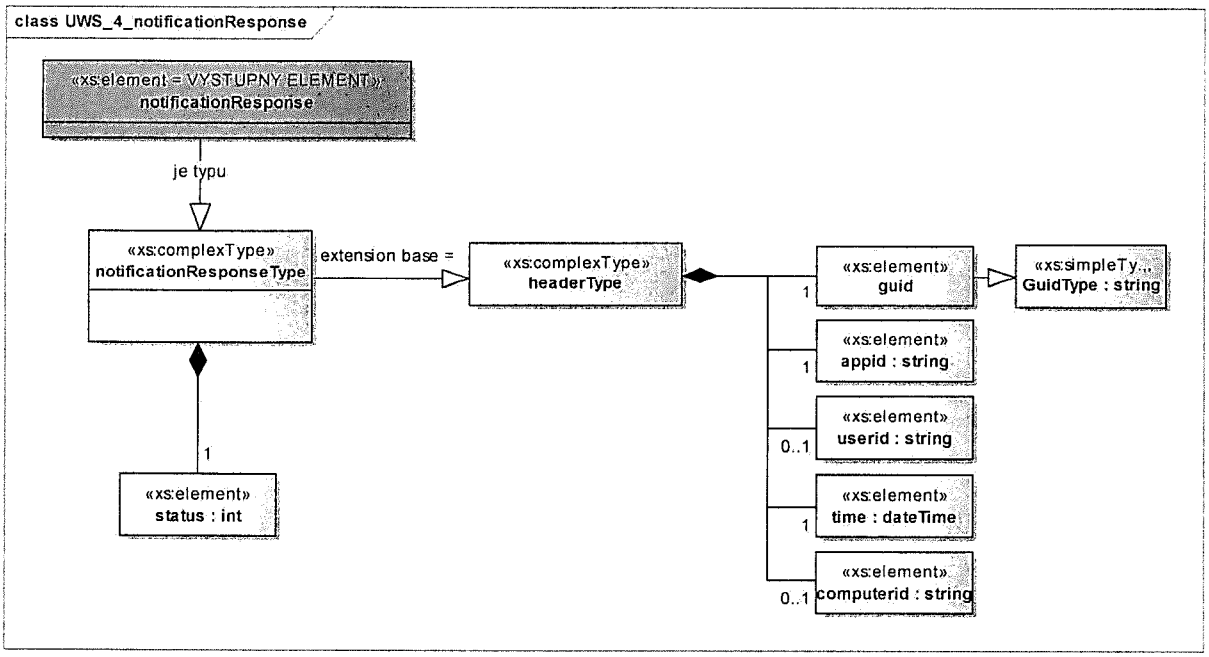

class UWS_3_notificationRequest
«xs:element = VSTUPNY ELEMENT»
notificationRequest
je typu
«xs:complexType»
notificationRequestType
extension base =
«xs:complexType»
HeaderType
«xs:element»
guid
«xs:simpleTy...
GuidType : string
«xs:element»
appid : string
«xs:element»
userid : string
«xs:element»
time : dateTime
«xs:element»
computerid : string
1
0..1
if notifySource = "Custom"
«xs:element»
notifySource
«xs:element»
notifySpecify
«xs:element»
transaction
«xs:simpleType...
NotifySourceType : string
«enum value»
- ObjChange
- Result
- Audit
- Custom
- CodeList
«xs:complexType»
ResultTransactionType
«xs:complexTy...
BasicTransaction
«xs:attribute»
- guid GUIDType
«xs:element»
operation
«xs:element»
txvariable
«xs:complexType»
ReferenceType
«xs:attribute»
- cooaddr :COOAddrType
- reference :COOAddrType
- mandant :COOAddrType
- explodeLevel xs:unsignedInt [0..5]
- keyClass :COOAddrType
- returnProps :ReturnPropsType = header
- definedProperties :xs:string
- definedHeaderAttrs :xs:string
«xs:simpleType»
COOAddrType : string
«xs:simpleType - en...
ReturnPropsType
«enum value»
- none
- header
- objclass
- defined
- values
- all
«xs:complexType»
TxVariableType
«xs:attribute»
- component :COOAddrType
- type :COOAddrType
- id xs:unsignedInt
«xs:element»
value
extension base =
«xs:complexType»
AssignmentType
«xs:attribute»
- index :xs:int
- guid :GUIDType
«xs:element»
value
«xs:complexType»
ObjectType
«xs:attribute»
- operation :OperationType = read
- fileName :xs:string
- lockKey :xs:string
- objName :xs:string
- objClass :COOAddrType
- notifyDiscard :xs:boolean
- lockRecursive :xs:boolean
- autoUnlock :xs:boolean
- lockTimeout :xs:int
- manualRefresh :xs:boolean
- notifyType :ObjectNotifyType
- searchLimit :xs:int
- foreignKey :xs:string
- mimeType :xs:string
- fileExport :xs:string
- checkUserAccess :COOAddrType
«xs:element»
property
«xs:complexType»
PropertyType
«xs:attribute»
- reference :COOAddrType
- count xs:unsignedInt
- initialize xs:boolean
- type mstns:COOAddrType
- name xs:string
«xs:simpleType - en...
OperationType
«enum value»
- create
- read
- modify
- search
«xs:simpleType - en...
ObjectNotifyType
«enum value»
- Create
- Modify
- Remove

4.2.2 Potvrdenie prevzatia notifikácie – notificationResponse

Výstupný element zasielaný externým systémom ako potvrdenie prevzatia notifikácie – `notificationResponse` je typu `notificationResponseType`, ktorý definuje jeho štruktúru:

- hlavička potvrdenia je definovaná typom `HeaderType` a obsahuje elementy `guid`, `appid`, `userid`, `time`, `computerid`;
- element `status`;

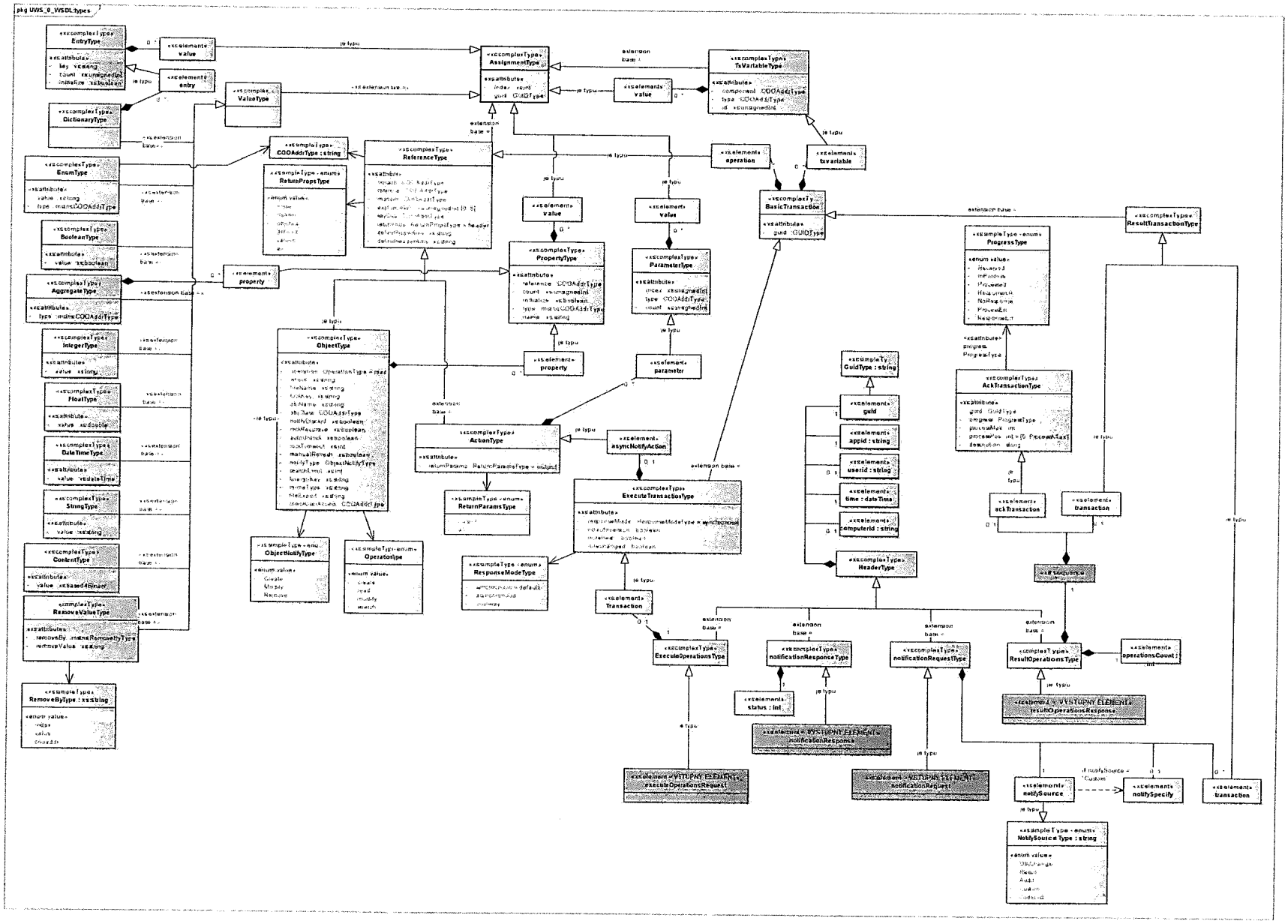
Nasledujúci obrázok obsahuje model tried popisujúci elementy, atribúty a dátové typy v rámci výstupného elementu `resultOperationsResponse`.



4.3 Elementy a typy údajov (types)

Na nasledujúcom obrázku sa nachádza diagram tried, ktorý popisuje všetky typy a elementy, použité v rámci UWS a UNS, a ich atribúty a vzťahy.

Poznámka: *Obrázok je do dokumentu vložený v dostatočnom rozlíšení. pre lepšiu čitateľnosť jednotlivých symbolov stačí zväčšiť jeho zobrazenie.*


pkg UWS_0_WSDL types
«xscomplexType» EntryType
«xselement» value
«xselement» entry
«xscomplexType» DictionaryType
«xscomplexType» EnumType
«xscomplexType» BooleanType
«xscomplexType» AggregateType
«xscomplexType» IntegerType
«xscomplexType» FloatType
«xscomplexType» DateTimeType
«xscomplexType» StringType
«xscomplexType» ContentType
«xscomplexType» RemoveValueType
«xscomplexType» RemoveByType : xs:string
«xscomplexType» ValueType
«xscomplexType» AssignmentType
«xscomplexType» TxVariableType
«xscomplexType» COOAddrType : string
«xscomplexType» ReferenceType
«xscomplexType - enum» ReturnPropsType
«xselement» property
«xselement» value
«xselement» operation
«xselement» txvariable
«xscomplexType» BasicTransaction
«xscomplexType» PropertyType
«xscomplexType» ParameterType
«xselement» parameter
«xscomplexType» ObjectType
«xscomplexType» ActionType
«xselement» asyncNotifyAction
«xscomplexType - enum» ReturnParamsType
«xscomplexType» ExecuteTransactionType
«xscomplexType - enum» ResponseModeType
«xselement» Transaction
«xscomplexType - enum» ObjectNotifyType
«xscomplexType - enum» OperationType
«xscomplexType» ExecuteOperationsType
«xselement - VYSTUPNY ELEMENT» executeOperationsRequest
«xscomplexType» notificationResponseType
«xselement» status : int
«xselement - VYSTUPNY ELEMENT» notificationResponse
«xscomplexType» notificationRequestType
«xselement - VYSTUPNY ELEMENT» notificationRequest
«xscomplexType» ResultTransactionType
«xscomplexType - enum» ProgressType
«xscomplexType» AckTransactionType
«xselement» ackTransaction
«xselement» transaction
«xsGroup»
«xscomplexType» ResultOperationsType
«xselement» operationsCount : int
«xselement - VYSTUPNY ELEMENT» resultOperationsResponse
«xselement» notifySource
«xselement» notifySpecify
«xselement» transaction
«xscomplexType - enum» NotifySource Type : string
«xsGuidType : string
«xselement» guid
«xselement» appId : string
«xselement» userId : string
«xselement» time : dateTime
«xselement» computerId : string
«xscomplexType» HeaderType

4.3.1 Hierarchia údajov

4.3.1.1 Priradenie – AssignmentType

Je základný abstraktný typ pre všetky typy hodnotových a referencovaných objektov. Umožňuje priradenie hodnoty alebo referencie do vlastnosti objektu, vlastnosti agregátu, položky slovníka a do parametra akcie.

Atribút `index` určuje polohu prvku pri vkladaní. Indexuje sa od 0. Pri pridávaní prvku na koniec poľa má index hodnotu `-1`. Pri vkladaní hodnoty do jednohodnotovej vlastnosti musí byť `index=0` alebo bez indexu.

V nasledujúcej tabuľke sú zobrazené korešpondujúce typy z XSD schémy a zo systému Fabasoft. Typy označené ako „V" sú hodnoty odvodené od `ValueType` a „R" sú referencované typy odvodené od `ReferenceType`.

Tabuľka 1 Typy hodnôt

Význam	XSD Typ	Typ*	Jedno hodnotový (Fabasoft)	Viac hodnotový (Fabasoft)
Logická hodnota	BooleanType	V	COOSYSTEM@1.1:BOOLEAN	COOSYSTEM@1.1:BOOLEANLIST
Celé číslo	IntegerType	V	COOSYSTEM@1.1: INTEGER	COOSYSTEM@1.1: INTEGERLIST
Desatinné číslo	FloatType	V	COOSYSTEM@1.1: FLOAT	COOSYSTEM@1.1: FLOATLIST
Enumerácia	EnumType*	V	Definované používateľom	Definované používateľom
Dátum a čas	DatetimeType	V	COOSYSTEM@1.1: DATETIME	COOSYSTEM@1.1: DATETIMELIST
Reťazec znakov	StringType	V	COOSYSTEM@1.1: STRING	COOSYSTEM@1.1: STRINGLIST
Obsah	ContentType	V	COOSYSTEM@1.1: CONTENT	COOSYSTEM@1.1: CONTENTLIST
Slovník	DictionaryType	V	COOSYSTEM@1.1: DICTIONARY	COOSYSTEM@1.1: DICTIONARYLIST
Agregát	AggregateType*	V	Definované používateľom	Definované používateľom
Objekt	ObjectType	R	COOSYSTEM@1.1: OBJECT	COOSYSTEM@1.1: OBJECTLIST
Akcia**	ActionType	R	---	---

* – referencia skutočného typu sa uvádza v atribúte;
** – akcia môže byť priradená ako hodnota len do zoznamu operácií;

4.3.1.2 Hodnota – ValueType

Abstraktný typ pre všetky typy jednoduchých aj zložených hodnôt. Odvodený je od typu `AssignmentType`. Neobsahuje žiadne vlastné atribúty.

4.3.2 Jednoduché typy hodnôt

Reprezentujú základné typy definované v systéme Fabasoft pre použitie v jednohodnotovej alebo viachodnotovej (list) reprezentácii vo vlastnostiach objektu alebo agregátu, v parametroch akcie a v položkách slovníka.

4.3.2.1 Logická hodnota – BooleanType

Obsahuje atribút `value` ktorý reprezentuje logickú hodnotu. V XML zápise sú hodnoty reprezentované ako „`true`" alebo „`false`".

Príklady zápisu v XML

Príklad zápisu v C#

```
BooleanType Logic = new BooleanType ();
Logic.Value = true;
```

Príklad zápisu v Jave

```
BooleanType Logic = new BooleanType ();
Logic.setValue(true);
```

4.3.2.2 Celé číslo – IntegerType

Obsahuje atribút `value` ktorý reprezentuje hodnotu celého čísla. Rozsah čísla je 32 alebo 64bit v závislosti od verzie systému.

Príklad zápisu

```
<value AB:type="AA:IntegerType" index="0" value="152" />
```

Príklad zápisu v C#

```
IntegerType Number = new IntegerType();
Number.Value = 152;
long N1 = Number.Value;
```

Príklad zápisu v Jave

```
IntegerType Number = new IntegerType();
Number.setValue(152);
int N1 = Number.getValue();
```

4.3.2.3 Enumerácia - EnumType

Obsahuje atribút `value` ktorý reprezentuje hodnotu enumerácie vo forme celého čísla. Rozsah čísla je 32 alebo 64bit v závislosti od verzie systému. V atribúte `type` je COO adresa typu enumerácie.

Príklad zápisu

```
value AB:type="AA:EnumType" index="0" value="1" />
```

Príklad zápisu v C#

```
EnumType Enum = new EnumType ();
Enum.Value = 1;
long N1 = Enum.Value;
```

4.3.2.4 Desatinné číslo – FloatType

Obsahuje atribút `value`, ktorý reprezentuje hodnotu vo forme čísla s pohyblivou rádovou čiarkou.

Príklad zápisu

```
value AB:type="AA:FloatType" index="0" value="0,5" />
```

4.3.2.5 Dátum a čas – DateTimeType

Obsahuje atribút `value`, ktorý reprezentuje hodnotu dátumu a času.

Príklad zápisu

```
value AB:type="AA:DateTimeType" index="0" value="2014-06-14T13:25:13" />
```

4.3.2.6 Reťazec znakov – StringType

Obsahuje atribút `value`, ktorý reprezentuje hodnotu reťazca znakov.

Príklad zápisu

[illegible]

4.3.2.7 Vymazaná hodnota - RemoveValueType

Špeciálny dátový typ pre odstránenie hodnoty bez ohľadu na typ hodnoty.

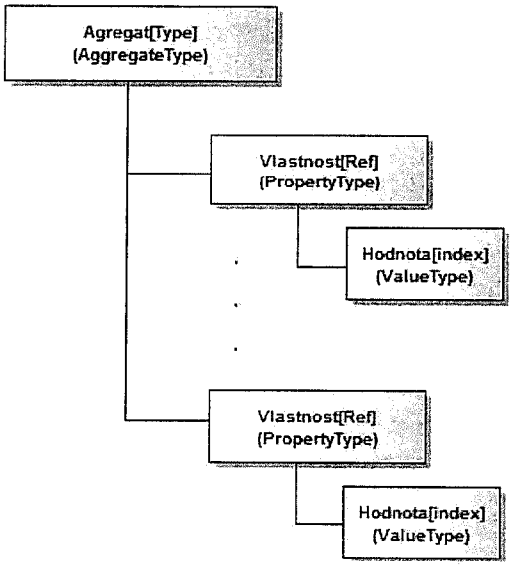
Atribút `removeBy` definuje, podľa čoho sa má hodnota vymazať a môže nadobúdať nasledovné hodnoty:

- `index` – vymazanie podľa pozície hodnoty vo viachodnotovej vlastnosti. V jednohodnotovej vlastnosti index=0; Vymaže sa hodnota na pozícii „`index`". Pre vymazanie položky na konci zoznamu `index=-1`.
- `value` – vymazanie podľa hodnoty – vymaže všetky položky, ktorých hodnota je rovná zadanej hodnote. Hodnota sa zadáva v atribúte `removeValue`;
- `cooaddr` – odstránenie ukazovateľa na objekt podľa COO adresy.

4.3.3 Agregát – AggregateType

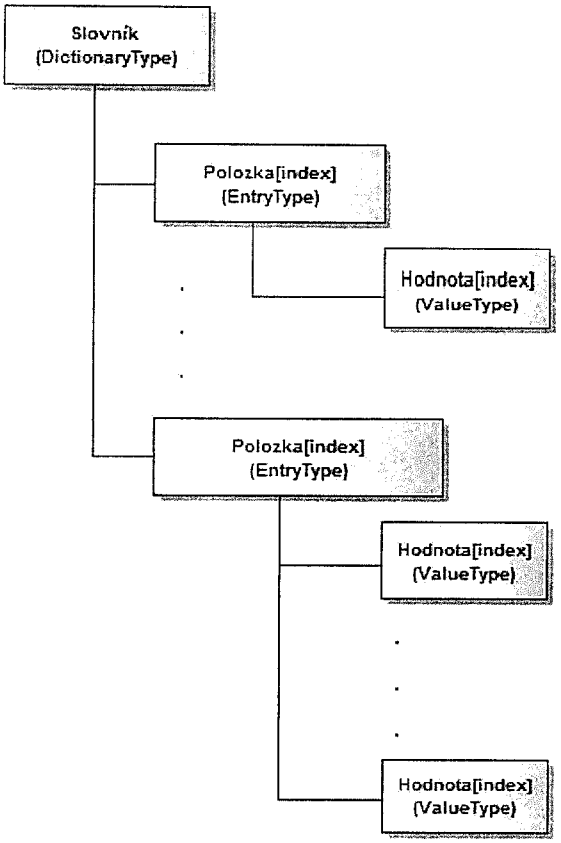
Agregát v systéme Fabasoft predstavuje zloženú vlastnosť, ktorá je obdobná ako štruktúra (struct) alebo rekord v klasických programovacích jazykoch.

Agregát obsahuje podľa typu uloženého v atribúte `type` množinu definovaných vlastností. Tieto sú uložené v elemente `property`, ktorý je typu `PropertyType`.



4.3.4 Slovník – DictionaryType

`DictionaryType` zodpovedá rovnomennej štruktúre v systéme Fabasoft, ktorá umožňuje ukladanie ľubovoľných heterogénnych štruktúrovaných a hierarchických informácii v čase behu systému.



Prvkom na ukladanie hodnôt do slovníka je element `entry`, ktorý je typu `EntryType`.

4.3.4.1 Položka slovníka – EntryType

Každá položka je identifikovaná kľúčom `key`. Pri napĺňaní položky hodnotami je možné vymazať všetky už existujúce hodnoty nastavením atribútu `initialize` na hodnotu `true`.

Hodnoty v položke sú uložené v elemente `value`, do ktorého je možné vkladať objekty `ObjectType` alebo hodnoty `ValueType`.

4.3.5 Obsah – ContentType

Obsah je typ hodnoty ktorý umožňuje uložiť alebo prečítať ľubovoľnú štruktúru a veľkosť údajov. Pre uloženie samotného obsahu slúži atribút `value`, ktorý je typu `base64Binary`.

Vzhľadom k používaným protokolom na prenos údajov je daný obsah prenášaný ako postupnosť `Base64`, čo zväčšuje jeho veľkosť a následne na cieľovom systéme je prekódovaný do pôvodného binárneho tvaru. Preto je nevhodné používať na prenos do alebo z vlastnosti s obsahom volanie webovej služby. Na tento účel je vhodnejšie používať WEBDav.

4.3.6 Transakčné premenné – TxVariableType

Tento typ umožňuje nastavenie transakčných premenných pre vykonanie transakcie, alebo vrátenie transakčných premenných, ak sú výsledkom transakcie. Transakčné premenné sa zapisujú do elementu `txvariable`.

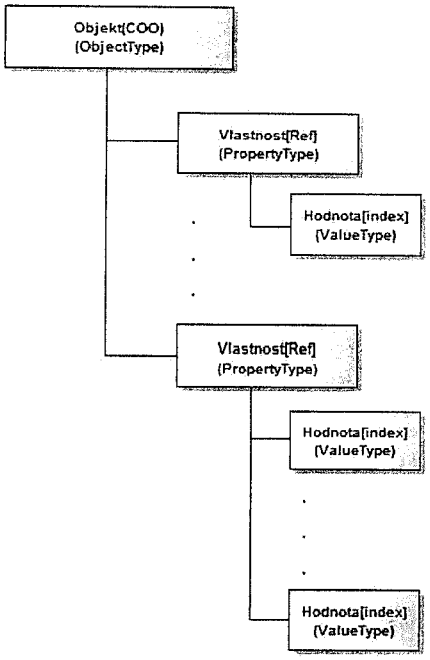
Atribúty typu `TxVariableType`:

- `component` – softvérový komponent transakčnej premennej.
- `type` – referencia typu parametra.
- `id` – identifikátor transakčnej premennej. Je jedinečný v rámci softvérového komponentu.

Element `value` slúži na zapísanie hodnoty premennej.

4.3.7 Objekt

Objekt je jeden zo základných prvkov a zodpovedá inštancii objektu v systéme Fabasoft. Objekt má svoju objektovú triedu (`objClass`), zoznam vlastností pre uloženie hodnôt (`property`) a akcie ktoré je možné nad objektom volať (operácia volania akcie).



4.3.7.1 Identifikácia objektu

Pri akejkoľvek práci s objektom je nutné objekt správne identifikovať. Systém používa dva druhy identifikácie lokálnu a globálnu.

Lokálna identifikácia

Slúži k identifikovaniu objektov vnútri zasielaného XML a na väzbu so zodpovedajúcimi objektmi v rámci volania služby a odpovede. Druhým prípadom je vytváranie objektu a jeho priraďovanie do iného objektu.

Pre lokálnu identifikáciu objektu v rámci volania, aj v prípade vytvárania objektu, slúži atribút `guid`. Tento atribút je typu `string` a mal by obsahovať jednoznačnú identifikáciu objektu v rámci volania. Odporúčame používať systémový generátor guid. Rovnaká hodnota tohto atribútu bude naplnená aj v zodpovedajúcom objekte v odpovedi volania, vďaka čomu je možné identifikovať aj novo vytvorený objekt.

Globálna identifikácia

Pre globálnu identifikáciu objektu v systéme Fabasoft, okrem prípadu vytvárania objektu, slúži atribút `cooaddr`, ktorý obsahuje COO adresu objektu.

4.3.7.2 Referencia na objekt – ObjectType

Používa sa pri vkladaní objektu ako hodnoty do vlastnosti iného objektu alebo agregátu, do parametra akcie alebo položky slovníka. Pre správne vloženie je nutné naplniť COO adresu do atribútu `cooaddr` a pri vkladaní do viachodnotovej vlastnosti použiť atribút `index`.

4.3.7.3 Vlastnosť – PropertyType

Zodpovedá vlastnosti objektu alebo agregátu v systéme Fabasoft. Obsahuje atribút `reference`, do ktorého musí byť vo vstupnom XML zapísaná referencia, ktorá danú vlastnosť jednoznačne identifikuje. Vo výstupnom XML atribút `reference` vypĺňa automaticky systém a na jeho základne možno identifikovať všetky vlastnosti.

Atribút `count` obsahuje vo výstupnom XML počet hodnôt vlastnosti. Tento atribút je duplicitný a počet vrátených hodnôt je možné zistiť aj z počtu vrátených elementov hodnoty (`value`). Vo vstupnom XML sa nepoužíva.

Atribút `initialize` pri nastavení na `true` slúži na odstránenie hodnôt vlastnosti pred napĺňaním nových hodnôt.

Vlastnosť obsahuje element `value`, ktorý je typu `AssignmentType`. V ňom sa prenášajú samotné hodnoty v rámci danej vlastnosti. Počet hodnôt, ktoré je možné zapisovať, závisí od toho, či je vlastnosť jednohodnotová alebo viachodnotová. Tieto informácie sú predmetom analytickej špecifikácie.

Atribút `type` – referencia typu vlastnosti.

Atribút `name` – meno vlastnosti tak ako je zobrazené v používateľskom rozhraní. Používa sa v odpovediach na volania WS.

4.3.8 Operácie nad objektom – ObjectType

Trieda `ObjectType` slúži na zápis jednotlivých operácií nad objektom vo vstupnom XML, na zápis objektov vrátených vo výstupnom XML a objektov zasielaných v rámci UNS.

Pre zápis operácií nad objektom vo vstupnom XML má tento typ niekoľko atribútov, ktoré sú spoločné pre všetky operácie, okrem výstupu a zápisu notifikácie. Spoločné vlastnosti sú popísané v nasledujúcich odsekoch, význam ostatných atribútov bude vysvetlený v samostatných kapitolách pre každú operáciu.

Pre všetky operácie (element `operation`) vo vstupnom XML musí byť definovaný atribút `operation` (typu `OperationType`), ktorý definuje typ vykonávanej operácie (`create`, `read`, `modify`, `search`). Význam jednotlivých typov operácií je popísaný v rámci samostatných kapitol. Vo výstupnom a notifikačnom XML sa tento atribút nevypĺňa.

Pre získanie požadovanej štruktúry údajov vo výstupnom XML sa definuje atribút `returnProps`, na základe ktorého sa vytvorí zoznam vlastností vrátených vo výstupnom XML. Môže nadobúdať nasledovné hodnoty:

- `none` – vo výstupnom XML sa objekt nezobrazí.
- `header` – objekt bude mať naplnené relevantné atribúty, element `property` bude prázdny.
- `objclass` – objekt bude mať naplnené relevantné atribúty, element `property` bude obsahovať vlastnosti definované v objektovej triede.
- `defined` – objekt bude mať naplnené relevantné atribúty, element `property` bude obsahovať vlastnosti, ktorých referencie boli zapísané v atribúte `definedProperties`. Referencie sa zapisujú oddelené bodkočiarkou.
- `values` – objekt bude mať naplnené relevantné atribúty, element `property` bude obsahovať vlastnosti, ktoré majú nenulovú hodnotu.
- `all` – objekt bude mať naplnené relevantné atribúty, element `property` bude obsahovať všetky vlastnosti objektovej triedy a aj všetky zdedené vlastnosti.

`definedHeaderAttrs` – definuje zoznam atribútov v hlavičke objektu (COO adresa, reference, objName, objClass), ktoré budú vrátené vo výstupnom XML. Ak je prázdny, budú vrátené všetky dostupné atribúty. Príklad: `definedHeaderAttrs` = "`cooaddr;objName`".

Vo výstupnom XML bude mať objekt vyplnené atribúty podľa nastavenia atribútu `returnProps` a typu objektu. V prípade, že `returnProps`= "`none`", nebude objekt vo výstupnom XML vôbec zobrazený. V ostatných prípadoch budú atribúty vyplnené podľa typu objektu nasledovne:

- `guid` – vždy keď bol atribút `guid` naplnený vo vstupnom XML.
- `cooaddr` – vždy, je to jednoznačná identifikácia objektu.
- `reference` – referencia objektu je naplnená pre komponentové objekty.
- `objName` – meno objektu je vyplnené pre každý objekt.
- `fileName` – je vyplnené pre objekty s obsahom.
- `objclass` – je vyplnený pre každý objekt.

- `foreignKey` – je vyplnený, ak má objekt cudzí kľúč (naplnenú vlastnosť `SKWEBSVC@AttrStrForeignKey`).

`explodeLevel` – nastavuje počet úrovní do ktorej sa objekty vo vlastnostiach riadia atribútom `returnProps`, pre objekt na poslednej úrovni `returnProps = "header"`. Ak atribút `explodeLevel` chýba, potom `explodeLevel = "0"`.

Štandardne je objekt spracovávaný (vytvorený, modifikovaný,...) v mandantovi alebo aktuálnej doméne, ktorú má používateľ, pod ktorým je WS volaná, štandardne predvolenú. V prípade potreby vykonania operácie s objektom v inom mandantovi musí byť atribút `mandant` (v rámci typu `ReferenceType`) naplnený COO adresou mandanta.

Ak je nad daným objektom povolená notifikácia zmien prostredníctvom univerzálnej notifikačnej služby, ktorá je v prípade konkrétneho volania nežiaduca, nastaví sa atribút `notifyDiscard` na `true`. Notifikácia bude vypnutá len pre daný objekt a v súčasne prebiehajúcej transakcii.

4.3.8.1 Vytvorenie objektu

Atribút `operation` v elemente `operation` musí mať hodnotu `create`. Pre vytvorenie objektu je nutné vyplniť atribút `objClass`, ktorý musí obsahovať referenciu objektovej triedy, inštanciu ktorej chceme vytvoriť. Zoznam objektových tried (referencií) a ich vlastností bude dodaný na analytickej úrovni.

Ak chceme pri vytvorení objektu naplniť aj obsah (týka sa to objektov s obsahom), názov súboru vložíme do atribútu `fileName`. Meno súboru musí obsahovať časť cesty od koreňového adresára uvedeného v konfigurácii používateľa pod ktorým sa uskutočňuje volanie. Ak je súbor vložený priamo v koreňovom adresári v atribúte sa uvedie len meno súboru.

V prípade, že potrebujeme vytvoriť inštanciu objektovej triedy podľa koncovky súboru, je možné v atribúte `objClass` uviesť namiesto triedy kľúčové slovo `ClassContentByExtension`. Systém nájde najvhodnejšiu triedu pre danú koncovku a v prípade, že koncovka nezodpovedá žiadnej triede, vytvorí objekt `Elektronický dokument` (`GENCONT@1.1:ContentObject`).

Pri vytváraní objektu je možné modifikovať vlastnosti rovnako ako pri modifikácii objektu. To znamená, že element `property` musí mať naplnené vlastnosti s príslušnými hodnotami, ktoré budú zmenené.

Tento proces prebehne po vytvorení objektu a načítaní obsahu (ak bol požadovaný).

4.3.8.2 Prečítanie objektu

Atribút `operation` v elemente `operation` musí mať hodnotu `read` a musí byť identifikovaný COO adresou, tzn. naplnený atribút `cooaddr`, prípadne `keyClass`, ak je objekt identifikovaný cudzím kľúčom.

Obsah elementu `property` je ignorovaný, zoznam vlastností pre výstup sa riadi atribútom `returnProps`, prípadne pre zoznam vrátených vlastností atribútom `definedProperties`.

4.3.8.3 Modifikácia objektu

Atribút `operation` v elemente `operation` musí mať hodnotu `modify` a musí byť identifikovaný COO adresou, tzn. naplnený atribút `cooaddr`, prípadne `keyClass`, ak je identifikovaný cudzím kľúčom.

Element `property` musí mať naplnené vlastnosti s príslušnými hodnotami, ktoré budú zmenené.

Zmena resp. naplnenie nového obsahu je možná vložením názvu súboru do atribútu `fileName`. Meno súboru musí obsahovať časť cesty od koreňového adresára uvedeného v konfigurácii používateľa, pod ktorým sa uskutočňuje volanie. Ak je súbor vložený priamo v koreňovom adresári, v atribúte sa uvedie len meno súboru.

4.3.8.4 Vyhľadanie objektu

Atribút `operation` v elemente `operation` musí mať hodnotu `search`.

Systém Fabasoft pracuje primárne s objektovým modelom dát ale samotné hodnoty ukladá do pripojeného relačného databázového systému, ktorý vo väčšine prípadov nekorešponduje s vlastnosťami objektových tried. Vyhľadávanie je preto komplikovanejšie vzhľadom k nutnosti konvertovať zadaný výber na jeden alebo niekoľko samostatných výberov nad DB tabuľkami a následne výsledky vyskladať do výsledného zoznamu nájdených objektov.

Pri zadávaní vyhľadávacích kritérií je potrebné zadať do atribútu `objClass` jednu alebo viac referencií objektových tried, v ktorých sa bude vyhľadávať. Pri viacerých triedach je oddeľovačom čiarka. Vyhľadávať sa bude v zadanej triede a všetkých odvodených triedach. Ak je potrebné vyhľadávanie iba v danej triede, do atribútu `where` je nutné pridať podmienku napr. `objClass = „COOSYSTEM@1.1:User“`.

Samotné vyhľadávacie kritérium sa zadáva do atribútu `where` a bližšie špecifikuje hodnoty pre vlastnosti z objektových tried zadaných v atribúte `objClass`. Ak je atribút `where` prázdny alebo sa nezadá, vyhľadajú sa všetky objekty v daných objektových triedach. V zadávaných podmienkach platia nasledovné pravidlá:

- Pre definovanie referencií vlastností používať vždy plnú referenciu. Pre všetky referencie bez softvérového komponentu bude doplnený komponent COOSYTSTEM@1.1. Je pravidlom začínať zápis referencie bodkou, ktorá znamená, že vlastnosť je priamo z objektu a nie je časťou zloženého typu.
- Pri adresovaní vlastností so zloženým typom sa používa zápis cesty k vlastnosti s bodkou ako oddeľovačom.
  Príklad: `where = ".COOMAPI@1.1:emailinformation.COOMAPI@1.1:mail“`
- Znakové reťazce sa definujú úvodzovkami alebo apostrofmi. Špeciálne znaky v reťazcoch sa prefixujú spätnou lomkou.
- Dátum a čas sa zadáva vo formáte: yyyy-mm-dd hh:mm:ss

V zápise podmienky možno použiť nasledovné kľúčové slová:

- `NOT` – logický operátor, invertuje hodnotu operandu.
- `AND` – logický operátor, výsledkom je logický súčin.
- `OR` – logický operátor, výsledkom je logický súčet.
- Operátory pre porovnávanie:
  - `<` – menší;
  - `<=` – menší alebo rovný;
  - `>` – väčší;
  - `>=` – väčší alebo rovný;
  - `=` – rovná sa;
  - `<>` – nerovná sa;
- `[SOUNDS] [NOT] LIKE` – určuje kedy sa ľavý reťazcový operand rovná pravému reťazcovému operandu. Znaky %,*,? a _ možno použiť ako zástupné znaky

v pravom operande. Pred operátorom `LIKE` možno použiť kľúčové slovo `SOUND` a operátor `NOT` pre fonetické porovnávanie.
Príklad: `where = "COOMAPI@1.1:emailinformation.COOMAPI@1.1:email LIKE '*fabasoft.com'"`

- [NOT] CONTAINS – trigger pre fulltextové vyhľadávanie.
  Príklad: `where = "COOSYSTEM@1.1:contcontent CONTAINS 'Workflow'"`
- [NOT] IN – určuje hodnoty z definovaného zoznamu.
- [NOT] INCLUDES – určuje, či hodnota pravého operandu je elementom v zozname ľavého operandu.
- [NOT] BETWEEN ... AND ... – určuje, či hodnota je v špecifikovanom rozmedzí.
- IS [NOT] NULL – určuje, či má vlastnosť nejakú hodnotu.
- UPPER – prevedie všetky znaky na veľké písmená (pre reťazcové dátové typy).
- LOWER – prevedie všetky znaky na malé písmená (pre reťazcové dátové typy).
- SUM – vypočíta súčet všetkých hodnôt vlastnosti (pre numerické dátové typy).
- AVG – vypočíta priemer všetkých hodnôt vlastnosti (pre numerické dátové typy).
- COUNT – vypočíta počet elementov vlastnosti.
- MIN – vyberie najmenšiu zo všetkých hodnôt vlastnosti (pre numerické dátové typy).
- MAX – vyberie najväčšiu zo všetkých hodnôt vlastnosti (pre numerické dátové typy).

Obmedzenie počtu vrátených objektov je možné naplnením maximálneho počtu vyhľadaných objektov do atribútu `searchLimit`.

4.3.8.5 Notifikácia zmeny objektu

Pri zasielaní notifikácie zmien v rámci univerzálnej notifikačnej služby je objekt vrátený obdobne ako pri odpovedi po operácii `read` alebo `modify`. Atribút `notifyType` (typu `ObjectNotifyType`) môže obsahovať nasledovné hodnoty:

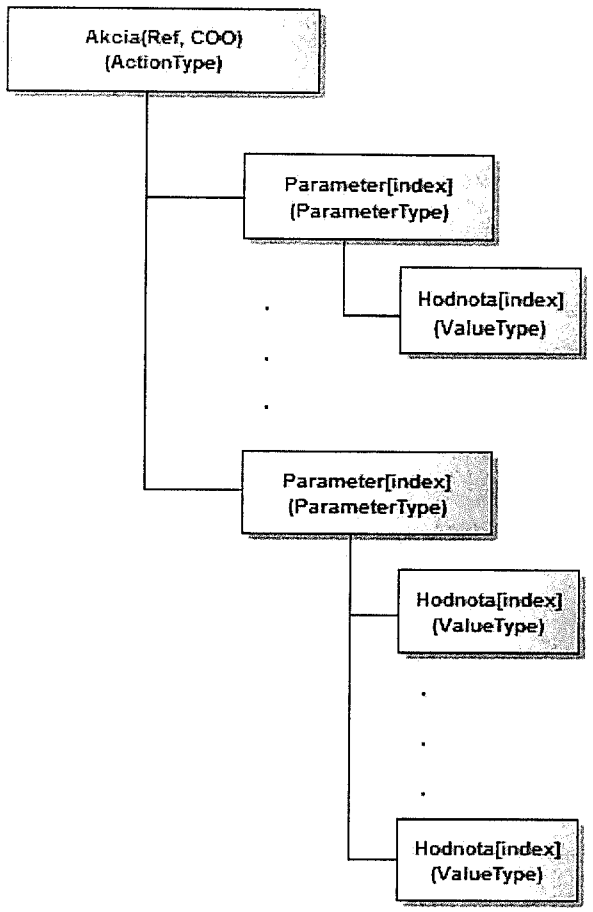
- `create` – objekt bol novo vytvorený a vlastnosti boli naplnené inicializačnými hodnotami.
- `modify` – objektu bola zmenená jedna alebo viac vlastností.
- `remove` – objekt bol zrušený (fyzicky zmazaný zo systému Fabasoft).

V prípade notifikácie sú v elemente property naplnené len tie vlastnosti, ktorým sa zmenila hodnota a sú dostupné pre notifikáciu.

Ostatné atribúty (`cooaddr`, `objName`,...) sú naplnené rovnako ako pri `returnProps = header`.

4.3.9 Operácia volanie akcie – ActionType

Akcia v systéme Fabasoft predstavuje ekvivalent metódy objektu v klasickom programovacom jazyku, rovnako má aj vstupné, výstupné a vstupno-výstupné parametre.



Pri volaní akcie je nutné vyplniť atribút `reference` referenciou akcie. V atribúte `cooaddr` je COO adresa objektu, nad ktorým sa akcia vykonáva, napr. COO adresa spisu. V prípade, kedy nie je potrebné volať akciu nad špecifickým objektom, možno naplniť atribút COO adresou alebo referenciou samotnej akcie.

Štandardne je akcia volaná (vykonávaná) v mandantovi alebo aktuálnej doméne, ktorú má používateľ, pod ktorým je WS volaná, štandardne predvolenú. V prípade potreby volania akcie v inom mandantovi musí byť atribút `mandant` naplnený COO adresou mandanta.

Pre identifikáciu akcie vo výstupnom XML pre načítanie parametrov je potrebné vyplniť atribút `guid`.

Na spresnenie zoznamu vrátených parametrov slúži atribút `returnParams`:

- `output` – vo výstupnom XML budú iba výstupné parametre (odporúčaná hodnota).
- `all` – vo výstupnom XML budú všetky parametre. Používa sa hlavne pri asynchrónnom spracovaní, kde výsledky spracovania asynchrónnej transakcie sa vracajú inému systému.

Parametre akcie je potrebné naplniť do elementu `parameter`.

4.3.9.1 Parameter akcie – ParameterType

Zodpovedá parametru akcie v systéme Fabasoft. Pri volaní musí byť vyplnený parameter `index`, ktorý identifikuje poradie parametra. Parametre sa číslujú od 1. Vo vstupnom XML nie je nutné vypĺňať všetky parametre, ale iba tie, ktorým chceme priradiť hodnotu.

Pre parametre, ktorým priraďujeme hodnotu, musíme priradiť aj jej typ podľa tabuľky 1 do atribútu `type`. Vo väčšine prípadov majú parametre akcie zadefinované jednoznačné typy, ale je možné použiť aj parameter bez typu a pri vykonávaní akcie sa riadiť podľa aktuálne použitého typu vloženej hodnoty.

Atribút `count` obsahuje vo výstupnom XML počet hodnôt parametra. Tento atribút je duplicitný a počet vrátených hodnôt je možné zistiť aj z počtu vrátených elementov hodnoty (`value`). Vo vstupnom XML sa nepoužíva.

5 Implementácia UWS a UNS na strane externých systémov

Aby bolo možné UWS a UNS použiť, je potrebné zrealizovať určitú implementáciu na strane externých systémov:

Univerzálna webová služba

UWS	WS vystavuje	Potrebná implementácia na strane EXTERNÉHO SYSTÉMU
Synchrónne spracovanie	**Fabasoft**	• Aplikačná logika pre generovanie a zasielanie vstupného volania (executeOperationsRequest) • a spracovanie došlej synchrónnej odpovede (resultOperationsResponse)
Asynchrónne spracovanie	**Fabasoft** - príjem vstupného volania	• Aplikačná logika pre generovanie a zasielanie vstupného volania (executeOperationsRequest) • a spracovanie synchrónneho potvrdenia (resultOperationsResponse)
	Externý systém - príjem asynchrónnej odpovede	• WS pre príjem a spracovanie asynchrónnej odpovede (notificationRequest), t.j. výsledku spracovania vstupného volania; • Aplikačná logika pre zasielanie potvrdenia asynchrónnej odpovede (notificationResponse)

Univerzálna notifikačná služba

UNS	WS vystavuje	Potrebná implementácia na strane EXTERNÉHO SYSTÉMU
Príjem notifikácií	**Externý systém** - príjem notifikácie	• WS pre príjem notifikačných volaní (notificationRequest); • Aplikačná logika pre synchrónne zasielanie potvrdenia prevzatia notifikácie (notificationResponse);

6 Prílohy

6.1 Zdroje

1. www.w3.org [online]. last updated: 2007-06-05 [cit. 2009-12-19]. Dostupná z WWW: < http://www.w3.org/TR/soap/>
2. www.w3.org [online]. last updated: 2001-03-14 [cit. 2009-12-19]. Dostupná z WWW: < http://www.w3.org/TR/wsdl>
3. www.w3.org [online] – http: <http://www.w3.org/Protocols/>

6.2 Tabuľka prístupových tried

V nasledujúcej tabuľke sa nachádza zoznam prístupových tried popísaných v kapitole Riadenie prístupových práv (ACL).

Názov prístupovej triedy	COO adresa prístupovej triedy
Zmeniť obsah(y)	COO.1.1.1.1911
Prečítať obsah(y)	COO.1.1.1.1910
Prečítať zoznam(y) objektov	COO.1.1.1.1909
Zmeniť zoznam(y) objektov	COO.1.1.1.341
Zmeniť vlastnosti	COO.1.1.1.339
Vyhľadať objekt	COO.1.1.1.338
Čítať vlastnosti	COO.1.1.1.336
Vytvoriť objekt	COO.1.1.1.335