. A s´emale´ır´o ´allom´any ´altal´anos szerkezete:

<?xml version="1.0" ?>

<xs:schema xmlns:xs="<http://www.w3.org/2001/XMLSchema>">

<!-- .... ide j¨on a s´ema r´eszletez´ese .... -->

</xs:schema>

A s´emafile szok´asos kiterjeszt´ese XSD. Mint l´athat´o a s´emale´ır´o XML doku-

mentum gy¨ok´ereleme a ’schema’ elem. A s´emal´ır´as elemeit a [’h](http://www.w3.org/2001/XMLSche)t[tp://www.w3.org/2001/XMLSche](http://www.w3.org/2001/XMLSche) n´evt´erbe tesszu¨k, ezzel jelezve a s´ema´ertelmez˝onek, hogy ezek az elemek

neki sz´olnak. A n´evt´er r¨ovid´ıt´esek´ent leggyakrabban az ’xs’ szimblumot haszn´alj´ak. Az ellen˝orzend˝o XML dokumentumban jelezni kell, hogy be kell olvasni a kijel¨olt s´emale´ır´o ´allom´anyt is. Ezt a dokumentum gy¨ok´erelem´en´el, speci´alis elemjellemz˝ok¨on keresztu¨l tehetju¨k meg:

<?xml version="1.0" ?>

<gyoker xmlns:xsi="[http://www.w3.org/2001/XMLSchema-instance"](http://www.w3.org/2001/XMLSchema-instance) xsi:SchemaLocation="file:/D:/kovi/jegyzetxml/minta1.xsd">

<!-- .... ide j¨on a dokumentum r´eszletez´ese .... -->

</gyoker>

# Az XMLSchema elemei

<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="<http://www.w3.org/2001/XMLSchema>"

-- alap´ertelemz´esi n´evterek --- >

<!-- k¨uls}o forr´asok bevon´asa -->

<!-- glob´alis elemek megad´asa -->

<!-- ¨osszetett t´ıpusok megad´asa -->

<!-- elemi t´ıpusok megad´asa -->

</xs:schema>

A glob´alis elemek azon elemeket jelentik, melyek a vizsg´alt XML doku- mentumban gy¨ok´erelemk´ent szerepelhetnek. Az XMLSchema szabv´anyban a s´ema t¨obb glob´alis elemet is tartalmazhat. A tervez˝onek c´elszeru˝ leszu˝k´ıteni a t´enylegesen haszn´alt gy¨ok´erelemekre.

Egy elem defini´al´asa az ELEMENT elemmel t¨ort´enik:

<xs:element jellemz}ok> tartalom </element>

A legfontosabb jellemz˝ok:

* name: elem azonos´ıt´o neve (sztring)
* type: elem t´ıpusa (sztring)
* ref: l´etez˝o glob´alis elemmel val´o egyez˝os´eget fejezi ki (elemn´ev)
* minOccurs: minim´alis el˝ofordul´assz´am (eg´esz)
* maxOccurs: maxim´alis el˝ofordul´assz´am (eg´esz)
* abstract: absztrakt elem jel¨ol´ese (true vagy false)

form: szu¨ks´eges-e a dokumentumban n´evt´er jel¨ol´es (qualified vagy un- qualified)

*•*

substitutionGroup: egym´assal felcser´elhet˝o elemek csomportj´anak ki- jel¨ol´ese (azonos´ıt´o n´ev)

*•*

Az elemn´el a ’type’ t´ıpuskijel¨ol´es is opcion´alis. Ha szerepel, akkor megad- hatjuk a elemt´ıpus azonos´ıt´oj´at. Ha nem, akkor egy n´evn´elku¨li t´ıpust is ki lehet jel¨olni. Mi most a n´evvel ell´atott t´ıpuskezel´esre helyezzu¨k a hangsu´lyt. Erre n´eh´any p´elda:

<xs:element name="ember" type="ember\_tipus"></xs:element>

<xs:element name="nev" type="xs:string"></xs:element>

A megadott t´ıpus lehet gy´ari alapt´ıpus ´es saj´at objektumt´ıpus is. A gy´ari t´ıpusok k¨ozu¨l itt csak n´eh´anyat sorolunk fel:

* string: sz¨oveges
* integer: eg´esz
* decimal: tizedes t¨ort
* double: duppla pontoss´agu´ t¨ort
* boolean: logikai
* date: d´atum
* time: id˝o
* ID: azonos´ıt´o DTD ´ertelemben
* IDREF: azonos´ıt´o hivatkoz´as DTD ´ertelemben

. Az elemet defini´al´o r´eszben szerepelhetnek a t´arolt ´ert´ekeket befoly´asol´o megk¨ot´esek. Ezen megk¨ot´eseket egy gyerek ellemmel ´ırjuk le a s´em´aban. Az ´ertelmezett megk¨ot´esek list´aja:

* key: els˝odleges kulcs
* keyref: idegen kulcs
* unique: egyedis´eg

A kulcs megad´as szintaktik´aja:

<xs:element name="nev" type="tipus" >

<xs:key name="megk¨ot´es neve" >

<xs:selector xpath="el´er´esi\_´ut1" />

<xs:field xpath="el´er´esi\_´ut2" />

</xs:key>

</xs:element>

A ’selector’ elemben szerep˝o ’el´er´esi u´t1’ jel¨oli azon elemek el´er´es´et (az XPath szabv´anynak megfelel˝o form´abban), amelyek az egyedek szimboliz´alj´ak, ezen elemeknek van egyedi ´ert´ekekkel b´ır´o tulajdons´aguk. Ezen tulajdons´agokat a ’field’ elemmel lehet megadni. Itt szint´en az XPath szabv´any szerinti le´ır´ast kell adni. Az el´er´esi u´tvonalakn´al a ’selector’ eset´eben a dokumen- tum gy¨oker´ehez kell viszon´ıtani, m´ıg a ’field’ eset´eben a selector elem lesz a relat´ıv gy¨ok´er. A k¨ovetkez˝o p´elda a dolgoz´okat nyilv´antart´o XML dokumen- tumban ´ırja el˝o a k´od mez˝o kulcs szerep´et. A s´ema´allom´any:

<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="<http://www.w3.org/2001/XMLSchema>" >

<xs:element name="fo" type="dolgozok\_tipus" >

<xs:key name="d\_k1">

<xs:selector xpath="dolgozo"></xs:selector>

<xs:field xpath="kod"></xs:field>

</xs:key>

</xs:element>

Az idegenkulcsok haszn´alat´ara a ’keyref’ integrit´asi elemet kell megadni a s´em´aban. A megk¨ot´es le´ır´as´anak alakja hasonl´o a kulcs defin´ıci´ohoz, a ku¨l¨ons´eg abban ´all, hogy itt szerepeletteni kell egy kulcs integrit´asi elemet is, melyb˝ol ´ert´eket vehet fel az idegenkulcs elem. A kapcsolt kulcselemet a kulcs megk¨ot´es azonos´ıt´o nev´evel jel¨olhetju¨k ki. A mintap´elda s´em´at ´es egy

´erv´enyes ´allom´anyt mutat be.

<?xml version="1.0" encoding="UTF-8"?>

<xs:schema xmlns:xs="<http://www.w3.org/2001/XMLSchema>”>

<xs:element name="fo" type="dolgozok\_tipus" >

<xs:key name="d\_k1">

<xs:selector xpath="./dolgozo"></xs:selector>

<xs:field xpath="kod"></xs:field>

</xs:key>

<xs:keyref refer="d\_k1" name="d\_k2">

<xs:selector xpath="./feladat"></xs:selector>

<xs:field xpath="felelos"></xs:field>

</xs:keyref>

</xs:element>

<xs:complexType name="dolgozok\_tipus">

<xs:sequence>

<xs:element name="dolgozo" maxOccurs="unbounded" type="dolgozo\_tipus” />

<xs:element name="feladat" maxOccurs="unbounded" type="feladat\_tipus” />

</xs:sequence>

</xs:complexType>

<xs:complexType name="feladat\_tipus">

<xs:sequence>

<xs:element name="leiras" type="xs:string"></xs:element>

<xs:element name="felelos" type="xs:integer"></xs:element>

</xs:sequence>

</xs:complexType>

<xs:complexType name="dolgozo\_tipus">

<xs:sequence>

<xs:element name="nev" type="xs:string"></xs:element>

<xs:element name="kod" type="xs:integer"></xs:element>

<xs:element name="fizetes" type="xs:integer"></xs:element>

</xs:sequence>

</xs:complexType>

</xs:schema>

A k¨ovetkez˝o elem az elemt´ıpu´sok meghat´aroz´as´ara szolg´al. A ’complex- Type’ ´es a ’simpleType’ parancsok szolg´alnak u´j elemt´ıpusok megad´as´ara. A ’simpleType’ eset´en csak sz¨ovegtartalma lehet az elemnek ´es nem tartozhat hozz´a jellemz˝o. Az ett˝ol elt´er˝o esetekben a ’complexType’ kulcssz´ot kell haszn´alni. Az ide tartoz´o parancsok alapform´aja:

<xs:simpleTpye name="t´ıpusn´ev" >

<!-- le´ır´as -->

</xs:simpleType>

<xs:complexTpye name="t´ıpusn´ev" >

<!-- le´ır´as -->

</xs:complexType>

Az ¨osszetett t´ıpus defini´al´asakor szerepelhet t¨obbek k¨oz¨ott egy ’mixed’ el- nevez´esu˝ jellemz˝o, melynek ’true’ ´ert´eke eset´en a t´ıpus m¨og¨ott vegyes modell

´all, azaz sz¨oveg ´es gyerekelem is el˝ofordulhat egyideju˝leg.

A megl´ev˝o elemi t´ıpusok szigor´ıt´asa eset´en a felvehet˝o ´ert´ekhalmazt korl´atozzuk:

<simpleType name="ujn´ev">

<restriction base="elemi\_t´ıpus" >

<!-- megszor´ıt´asok -->

</restriction>

</simpleType>

A megszor´ıt´asok k¨oz¨ott az al´abbi lehet˝os´egek ´elnek:

* minExclusive: minim´alis ´ert´ek megad´asa (nyilt v´eg)
* maxExclusive: maxim´alis ´ert´ek megad´asa (ny´ılt v´eg)
* minInclusive: minim´alis ´ert´ek megad´asa (z´art v´eg)
* maxInclusive: maxim´alis ´ert´ek megad´asa (z´art v´eg)
* totalDigits: sz´amokn´al a sz´amjegypoz´ıci´ok darabsz´ama
* fractionDigits: sz´amokn´al a t¨ortr´eszt le´ır´o sz´amjegyek darabsz´ama
* length: sz¨oveg, lista pontos hossza
* minLength: sz¨oveg minim´alis hossza
* maxLength: sz¨oveg maxim´alis hossza
* enumeration: felvehet˝o ´ert´ekek list´aja
* pattern: sz¨oveg form´atum´at el˝o´ır´o regul´aris kifejez´es P´eldak´ent vegyu¨k egy ´eletkor t´ıpus l´etrehoz´as´at:

<simpleType name="eletkor">

<restriction base="integer">

<minInclusive value="0" />

<maxInclusive value="130" />

</restriction>

</simpleType>

Az aut´ok rendsz´amainak t´arol´as´ara szolg´al´o t´ıpust regul´aris kifejez´essel lehet megadni:

<simpleType name="rendszam\_tipus">

<restriction base="string">

<pattern value="[A-Z]{3}-\d{3}" />

</restriction>

</simpleType>

Komplex t´ıpusok eset´en lehet˝os´eg van a gyermek elemek struktu´r´aj´anak megad´as´ara. Az XMLSchema az al´abbi konstrukci´os m´odszereket t´amogatja:

* sequence: elemek szekvenci´alis megad´asa, r¨og´ıtett sorrend
* choice: a megadott elemekb˝ol egyet lehet kiv´alasztani felhaszn´al´asra

all: a megadott elemek maximum egyszer fordulhatnak el˝o, de tetsz˝oleges sorendben k¨ovetkezhetnek az elemek

*•*

Az egyes konstrukci´os modulok egym´asba ´agyazhat´ok, mint azt a k¨ovetkez˝o p´elda is mutatja:

<complexType name="rendeles\_tipus">

<sequence>

<element name="vevo" type="string"/>

<choice>

<element name="postai\_cim" type="cim\_tipus"/>

<element name="szamlaszam" type="szamlaszam\_tipus"/>

</choice>

<element name="termek" type="termek\_tipus" maxOccurs="unbounded" />

<all>

<element name="osszeg" type="integer" />

<element name="datum"n type="dateTime" />

</all>

</sequence>

</complexType>

Az ¨osszetett t´ıpusokn´al az elemek mellett elemjellemz˝ok is megjelenhet- nek. Az elemjellemz˝o hozz´arendel´ese a t´ıpushoz a

<attribute name="jellemzo\_neve" type="t´ıpus" param´eterek />

s´emaelemmel t¨ort´enik. Az itt el˝ofordul´o param´eterek k¨ozu¨l a legfontosabbak: use: megadja, hogy k¨otelez˝o vagy opcion´alis a jellemz˝o; lehets´eges

*•*

´ert´ekek: required, optional vagy prohibited

* fixed: a jellemz˝o r¨ogt´ıtett ´ert´eku˝
* default: a jellemz˝o alap´ertelmez´esi ´ert´eke

P´eldak´ent vegyu¨k azt az esetet, amikor a rendel´eshez egy k¨otelez˝o ren- del´esi sz´amot kapcsolunk:

<complexType name="rendeles\_tipus">

<sequence>

<element name="vevo" type="string"/>

<choice>

<element name="postai\_cim" type="cim\_tipus"/>

<element name="szamlaszam" type="szamlaszam\_tipus"/>

</choice>

<element name="termek" type="termek\_tipus" maxOccurs="unbounded" />

<all>

<element name="osszeg" type="integer" />

<element name="datum" type="dateTime" />

</all>

</sequence>

<attribute name="rend\_szam" type="string" use="required" />

</complexType>

. Az XMLSchema ´ertelmez´es szerint ¨osszetett t´ıpus az is, amikor ugyan sz¨oveges ´ert´ek szerepel, de jellemz˝o kapcsol´odik az elemhez. A s´emale´ır´o nyelvben az elemi, sz¨oveges tartalom megad´as´ara a ’simpleCon- tent’ elem szolg´al. , mely mint egy keretk´ent lehet˝os´eget ad az ig´enyelt elemi t´ıpus defini´al´asra a megl´ev˝o alapt´ıpusokra ´ep´ıtve. A sz´armaztat´asnak k´et m´odja van, a leszu˝k´ıt´es ´es a kiterjeszt´es. Ezen mu˝veletekre a ’restriction’ ´es az ’extension’ elemek szolg´alnak. A kiterjeszt´es u´j elemjellemz˝ok hozz´aad´as´at teszi lehet˝ov´e. P´eld´ak´ent vegyu¨nk egy ´arat le´ır´o elemet, melyben az ´ert´ek numerikus eg´esz, ´es rendelkezik egy p´enznem jellemz˝ovel:

<xs:complexType name="ar\_tipus" >

<xs:simpleContent>

<xs:extension base="xs:integer">

<xs:attribute name="penznem" type="xs:integer">

</xs:attribute>

</xs:extension>

</xs:simpleContent>

</xs:complexType>

A kiterjeszt´esn´el a b´azist´ıpus lehet elemi t´ıpus vagy olyan ¨osszett t´ıpus, melyben a tartalom elemi, sz¨oveges.

Az ¨osszetett tartalmat a ’complexType’ elemmel jel¨olju¨k. P´eld´ankban egy megl´ev˝o ’dol- gozo tipus’-b´ol sz´armaztatunk egy ’fonok tipust’ egy ’dolgozo letszam’ gy- erekelem hozz´aad´as´aval:

<xs:complexType name="dolgozo\_tipus">

<xs:sequence>

<xs:element name="nev" type="xs:string"></xs:element>

<xs:element name="kod" type="xs:integer"></xs:element>

<xs:element name="fizetes" type="xs:integer"></xs:element>

</xs:sequence>

</xs:complexType>

<xs:complexType name="fonok\_tipus">

<xs:complexContent>

<xs:extension base="dolgozo\_tipus">

<xs:sequence>

<xs:element name="dolg\_letszam" type="xs:integer"></xs:element>

</xs:sequence>

</xs:extension>

</xs:complexContent>

</xs:complexType>

O¨ sszetett t´ıpus eset´en a kiterjeszt´eben az al´abbi konstruktor elemek szere- pelhetnek:

* sequence: elemek k¨otelez˝o list´aja
* choice: elemek v´alaszt´asi list´aja, egy keru¨l kiv´alaszt´asra
* all: opcion´alis elemek tetsz˝oleges sorrendben
* attribute: elemjellemz˝o

anyAttribute: megengedi tetsz˝oleges, a s´em´aban nem defini´alt elem- jellemz˝o haszn´alat´at

*•*

* attributeGroup: elemjellemz˝ok csoportja
* Group: elemek csoportja
* annotation: megjegyz´es

A megszor´ıt´as (restriction) eset´en a felvehet˝o ´ert´ekhalmazt korl´atozzuk A k¨ovetkez˝o p´eld´aban a ’dol- gozo tipust’ olyan ’fonok tipus’-ra alak´ıtjuk ´at, melyben a fizet´es ´ert´eke le- gal´abb 20000 egys´eg.

<xs:complexType name="fonok\_tipus">

<xs:complexContent>

<xs:restriction base="dolgozo\_tipus">

<xs:sequence>

<xs:element name="nev" type="xs:string"></xs:element>

<xs:element name="kod" type="xs:integer"></xs:element>

<xs:element name="fizetes" >

<xs:simpleType>

<xs:restriction base="xs:integer">

<xs:minInclusive value="20000"></xs:minInc

</xs:restriction>

</xs:simpleType>

</xs:element>

</xs:sequence>

</xs:restriction>

</xs:complexContent>

</xs:complexType>

Mint l´athat´o, a szu˝k´ıt´es u´gy oldhat´o meg, hogy a m´odos´ıtand´o elemn´el nem egy gy´ari alapt´ıpust adunk meg, hanem annak egy lesz´armazt´as´at a ’simpleContent’ mechanizmus keret´eben.

Ku¨l¨on ´erdemes megjegyezni azt az esetet, amikor egy u¨res ´es jellemz˝okkel rendelkez˝o elemre van szu¨ks´eg, melyet p´eld´aul az al´abbi minta is lefed:

<ar ertek="2300" penznem="Ft"/>

Az ide vonatkoz´o t´ıpust ¨osszetett t´ıpusk´ent kell defini´alni, melyhez nem tar- tozik t´enyleges tartalom. A b´azist´ıpus szerep´et most a speci´ais jelent´esu˝ ’any- Type’ szimb´olum t¨olti be, amely a s´ema ¨osszes t´ıpus´anak ˝ost´ıpus´at jel¨oli. A sz´armaztat´as a megszor´ıt´as mu˝velet´en alapszik, s csak azt deifni´aljuk amire szu¨ks´eg lesz.

<xs:complexType name="ujar\_tipus">

<xs:complexContent>

<xs:restriction base="xs:anyType">

<xs:attribute name="ertek" type="xs:integer"/>

<xs:attribute name="penznem" type="xs:string"/>

</xs:restriction>

</xs:complexContent>

</xs:complexType>

<xs:element name="ar" type="ujar\_tipus"/>