**DTD**

A XML szerkezet´enek ellen˝orz´es´ere szolg´al´o egyik legelterjedtebb m´odszer a DTD (Document Type Definition) mechanizmus. A DTD sz- erepe az XML dokumentumok szerkezet´enek korl´atoz´asa. A DTD le´ır´as egy s´emale´ır´asnak tekinthet˝o. Ez a s´emale´ır´as a rel´aci´os s´em´ahoz hasonl´o sz- erepet t¨olt be, de ann´al j´ovabb laz´abb szereppel b´ır. Egy DTD le´ır´asnak rendszerint t¨obb XML dokumentum is megfelel. Egy XML dokumentumot validaltnak, ellen˝orz¨ottnek nevezu¨nk, ha az teljes´ıti a megadott DTD (vagy m´as s´emal´ır´o) k¨ovetelm´enyiet.

A DTD eset´eben a s´ema le´ır´asa t¨ort´enhet az XML dokumentum elej´en ´es egy ku¨l¨on s´emal´ır´o ´allom´anyban is. A s´ema legfontosabb c´elja, hogy megadja

* milyen elemek ´ertelmezettek
* milyen elemjellemz˝ok haszn´alhat´ok
* milyen az egyes elemek bels˝o szerkezete (befoglalt elemek)
* az alkot´o elemek sz´amoss´ag korl´atoz´asai
* ´ert´ekkorl´atoz´asok bevezet´ese
* egyszeru˝s´ıt˝o szimb´olumok defin´ı´al´asa

# A DTD szintaktik´aja

A DTD le´ır´asra utal´o hivatkoz´asnak a viszg´aland´o XML dokumentum elej´en kell elhelyezkednie. A hivatkoz´ast egy megjegyz´es jellegu˝ elemben helyezzu¨k el. Ha a s´emale´ır´as egy ku¨ls˝o a´llom´anyban van, akkor a hivatkoz´as alakja:

<!DOCTYPE gy¨ok´erelem\_neve SYSTEM "k¨uls¨o ´allom´any neve"’>

Ha az XML dokumentum belsej´eben adjuk meg a szerketetre vomnatkoz´o megk¨ot´est, akkor ahhoz a

<!DOCTYPE gy¨ok´erelem\_neve [ s´ema\_le´ır´as ]>

formatum haszn´aljuk.

A bels˝o s´emal´ır´asban az al´abbi szerkezeti komonensek szerepelhetnek:

* elem
* elem jellemz˝o
* egyed szimb´olumok

Az elem megad´asakor megadjuk a nev´et ´es a bels˝o szerkezet´et, s´em´aj´at:

<!ELEMENT elem\_n´ev szerkezet>

A szerkezet megad´as´anak f˝obb t´ıpusai:

* EMPTY : u¨res elem, egytagu´ tartalom elem
* (#PCDATA) : sz¨oveg ´ert´eket tartalmaz´o elem
* ANY : tetsz˝oleges tartalom megengedett
* (szerkezet) : gyerekelemeket tartalmaz´o elem (ELEMENT-ONLY t´ıpus)
* (#PCDATA | szerekezet)\* : vegyes, sz¨oveget ´es gyerek elemeket is

tartalmaz´o elem (MIXED t´ıpus). U¨ gyelni kell arra, hogy ebben az

esetben csak az itt megadott form´atumot fogadja el az ´ertelmez˝o, nem lehet p´eld´aul a \* sem elhagyni.

Tov´abb r´eszletezve a szintaktik´at, az elem-szerkezet megad´asa k¨ovetkezik.

Itt kellmegadni, hogy milyen t´ıpusu´ gyerekelemek fordulhatnak el˝o, s mi-

lyen lehet azok el˝ofordul´asi sorrendje. Ehhez egyr´eszt sz´amoss´ag jelz˝o sz- imb´olumokat haszn´alunk a szerkezeti csoport v´eg´en, m´asr´eszt a csoporton belu¨li el˝ofordul´asok is tehet˝ok megk¨ot´esek. A sz´amoss´agjelz˝ok ´ert´ekei:

* ()\* : tetsz˝oleges sz´amu´ el˝ofordul´as
* ()+ : egy vagy t¨obb el˝oforul´as
* ()? : nulla vagy egy el˝ofordul´as
* () : pontosan egy el˝ofordul´as

A csoporton belu¨li viszonyt megad´o elemek:

* *e*1*, e*2 : az elemek szekvenci´aja
* *e*1|*e*2 : az egyik elem fordul el˝o a k´et elemb˝ol

Az elemjellemz˝ok eset´en az a´ltal´anos formula alakja a k¨ovetkez˝o:

<!ATTLIST elem jellemz¨o\_neve tipus alap´ert´ek>

A parancsban az elem a befoglal´o elem neve. A t´ıpus a jellemz˝o ´ert´ek´enek jelleg´et mutatja. Itt nem lehet programoz´asi t´ıpusszinten pontos´ıtani az

elemhet tartoz´o ´ert´eket (teh´at nem lehet megadni, hogy d´atum t´ıpusu´ a

jellemz˝o). Csak az ´ert´ek jelleg´et hat´arozhatjuk meg. Az ´ertelmezett jelleg t´ıpusok:

* CDATA : sz¨oveges ´ert´ek
* (*e*1|*e*2) : ´ert´eklista

ID : egyedi azonos´ıt´o ´ert´eket tartalmaz a jellemz˝o, hasonl´ıt a kulcs integrit´asi megk¨ot´eshez

•

IDREF : egy ID ´ert´ek´et tartalmazza, hasonl´ıt az idegen kulcs integrit´asi megk¨ot´eshez

•

* IDREFS : t¨obb ID ´ert´ekeit tartalmazza

NMTOKEN : a jellemz˝o egy XML azonbos´ıt´o n´evet tartalmaz, teh´at nem kezd˝odhet pl. sz´ammal a t´arolt ´ert´ek

•

* ENTITY : egy egyed szimb´olum ´ert´ek´et t´arolja a jellemz˝o

NOTATION : egy ku¨ls˝o objektumra utal´ast tartalmaz az ´ert´ek (ez lehet p´eld´aul egy ku¨ls˝o feldolgoz´o program azonos´ıt´oja)

•

Az ID t´ıpusu´ elemjellemz˝o kulcs szerep´et mutatja az is, hogy egy elemn´el csak egyetlen ID t´ıpusu´ jellemz˝o szerepelhet.

Az alap´ert´eket megad´o param´eterr´esz azt szab´alyozza, hogy az elem- jellemz˝onek k¨otelez˝o-e ´ert´ekkel rendelkeznie vagy sem. A param´eter lehets´eges

´ert´ekei:

* #REQUIRED : k¨otelez˝o explicit ´ert´eket tartalmaznia

\* #FIXED ´ert´ek : egy r¨ogz´ıtett ´ert´ekkel rendelkezik a jellemz˝o. A r¨ogz´ıtett ´ert´eket a FIXED kulcssz´o ut´an kell megadni.

•

* #IMPLIED : nem k¨otelez˝o ´ert´eket megadni

´ert´ek : az alap´ertelmez´esi ´ert´ek kijel¨ols´e´ere szolg´al, ilyenkor az IM- PLIED t´ıpust fogja haszn´alni az ´ertelmez˝o

•

Az elemek ´es elemjellemz˝ok mellett szerepeltethet˝ok m´eg a DTD le´ır´asban a ritk´abban haszn´alatos egyed-szimb´olumok. Az egyed szimb´olumok valamely hosszabb sz¨ovegr´esz r¨ovid´ıt´es´ere, alias n´evvel val´o ell´at´as´ara szolg´al. Az egyed-szimb´olumok defini´al´asa t¨ort´enhet bels˝o ´es ku¨ls˝o forr´asb´ol. A ku¨ls˝o forr´as eset´en egy ku¨ls˝o a´llom´anyb´ol keru¨l beolvas´asra a helyettes´ıt´esi ´ert´ek. Bels˝o defin´ıci´o eset´en az XML ´allom´anyban adjuk meg a szimb´olumhoz tar- toz´o ´ert´eket. A bels˝o defin´ıci´o alakja:

<!ENTITY szimb´olum "´ert´ek">

A kapcsol´od´o XML dokumentumban az ´ert´ek helyett haszn´alhatjuk a

< > ... &szimb´olum; ... < >

r¨ovid´ıt´est. Ku¨ls˝o defin´ıci´o eset´en az ´ert´ek hely´en a tartalmaz´o le´ır´o a´llom´any szerepel:

<!ENTITY szimb´olum SYSTEM "´allom´any">

A fenti egyedszimb´olumok mellett a DTD szabv´any t´amogatja az u´gynevezett param´eter szimb´olumok haszn´alat´at is. A param´eter szimb´olom olyan sz- imb´olum, amelyet a DTD defin´ıci´os r´eszben lehet haszn´alni. Vagyis ezzel

a mechanizmussal param´eteres s´ema defin´ıci´ot lehet v´egrehajtani. A s´ema defin´ıci´oban egy szimb´olum szerepelhet, s annak aktu´alis tartalam´at´ol fu¨gg az

´erv´enyesnek tekintett s´ema. A param´eter szimb´olum l´etrehoz´as´anak parancsa:

<!ENTITY  szimb´olum "´ert´ek">

<!ENTITY  szimb´olum SYSTEM "´allom´any">

A l´etrehozott szimb´olumot a DTD defin´ıci´os r´eszben lehet felhaszn´alni a k¨ovetkez˝o m´odon:

<!DOCTYPE ... [

szimb´olum;

>

Alap´ertelmez´esben a szimb´olumegy teljes elemet vagy jellemz˝olist´at tar- talmazhat. Erre p´elda az al´abbi kis minta:

<!DOCTYPE fo [

<!ENTITY elem2 "<!ELEMENT al1 EMPTY>

<!ELEMENT fo (#PCDATA | al1)\* >

elem2;

>

A fenti mechanizmus el˝onye, hogy seg´ıts´eg´evel ku¨ls˝o a´llom´anyokb´ol egyes elemek ku¨l¨on is a´temelhet˝ok. Ekkor csak az ig´enyelt elemek defin´ıci´oja j¨on a´t, nem szu¨ks´eges a teljes defin´ıci´os a´llom´any alkalmaz´asa.

A param´eter szimb´olumok m´asik lehets´eges alkalmaz´asi teru¨lete a felt´eteles defin´ıci´o l´etrehoz´esa. Egyes ´ertelmez˝ok t´amogatj´ak a felt´eteles ´ertelmez´esi szekci´ok haszn´alat´at, azaz a defin´ıci´o egyes r´eszei kihagyhat´ok vagy bevon- hat´ok a s´ema fel´ep´ıts´ebe. A kihagyand´o r´esz megad´asa:

<![IGNORE[

s´ema defin´ıci´o

]]>

A bevonand´o r´esz behat´arol´asa:

<![INCLUDE[

s´ema defin´ıci´o

]]>

A param´eter szimb´olumot ekkor arra lehet haszn´alni, hogy kijel¨olju¨k, aktualisan mely r´eszt kell bevonni vagy elhagyni. Az ide vonatkoz´o s´emar´eszlet:

<!ENTITY  konyv "INCLUDE" >

<!ENTITY  cd "IGNORE" >

<![konyv;[

<ELEMENT konyv (cim, kiado,..)>

<ATTLIST konyv isbn ID #REQUIRED>

...

]]>

<![cd;[

<ELEMENT cd (cim, kiado,..)>

<ATTLIST konyv cdid ID #REQUIRED>

...

]]>