RAPORTARE ŞTIINŢIFICĂ

Proiect complex ReTeRom. Proiect component CoBiLiRo

Activitatea A2.2: ***Raport asupra soluţiilor de armonizare a reprezentărilor colecţiilor existente text /vorbire (metadate şi adnotări)***

Faza de predare: iunie 2019

Autori: Ionuţ Pistol, Diana Trandabăţ, Andrei Scutelnicu, Cristian Pădurariu
Colaborator extern: Şerban Boghiu

1. Rezumatul etapei

A doua etapă (2019) a proiectului CoBiLiRO prevede realizarea infrastructurii pentru gestionarea şi stocarea corpusului bimodal. Activităţile complementare prevăzute includ descrierea şi implementarea unor soluţii de armonizare a reprezentărilor colecţiilor existente text/vorbire (metadata şi adnotări). Ulterior, aceste soluţii urmează să fie utilizate pentru a armoniza formatele colecţiilor existente şi a le încărca pe platforma CoBiLiRO. Similar celorlalte etape, este prevăzută o activitate de diseminare atât la evenimente ştiinţifice cât şi în mass-media. De interes special pentru platforma dezvoltată este respectarea drepturilor de autor şi a anonimizării solicitate pentru contributorii de resurse pe platformă.

2. Rezumatul activităţii

Activitatea 2.2 are ca obiectiv descrierea unor soluţii de conversie automată a celor trei formate identificate ca fiind utilizate de contributorii actuali la formatul CoBiLiRO, descris în raportul activităţii A1.3. Conversia are ca scop facilitarea mecanismelor de căutare şi comparare a resurselor contribuite pe platforma proiectului, precum şi facilitarea utilizării formatului propus ca standard pentru resursele aliniate text-voce. Pe viitor, în cazul în care noi formate vor fi identificate, vor putea fi propuse procese de conversie similare.

3. Descrierea ştiinţifică şi tehnică
3.1 Introducere

În etapa 2018 a proiectului, două dintre activităţile finalizate în cadrul CoBiLiRO au fost A1.2. *Inventarierea colecţiilor de date lingvistice româneşti disponibile la parteneri sau în terţe coaliţii* şi A1.3. *Proiectarea funcţională şi arhitecturală a infrastructurii care va găzdui resursele şi instrumentele de prelucrare şi acces ale consorţiului şi realizarea unui prototip*,

ambele fiind descrise în raportul etapei anterioare. În urma acestor activităţi au fost identificate trei formate predominant utilizate în reprezentarea resurselor partenerilor. De asemenea, au fost identificate cerinţele platformei implementate în această etapă şi a fost descris un format de reprezentare ce va fi folosit pentru punerea la dispoziţie către utilizatori a resurselor contribuite. Pentru descrierea detaliată a formatului agreat de parteneri, ce va fi folosit pentru stocarea resurselor pe platforma CoBiLiRO, poate fi consultat raportul corespunzător activităţii A1.3 [3] aferent etapei 2018. În acest raport, acest format va fi referit ca formatul CoBiLiRo.

Acest raport continuă în Secţiunea 2 cu o descriere pe scurt a celor trei formate menţionate. Secţiunea 3 propune soluţii de armonizare pentru conversia acestor trei formate de intrare la formatul platformei. Procesele de conversie propuse ţin cont şi de eventualele incompatibilităţi sau omisiuni ale formatelor originale. Secţiunea 4 a raportului discută şi posibile optimizări necesare pe serverul platformei CoBiLiRO în vederea facilitării conversiei automate a unui volum mare de fişiere.

3.2 Formatele folosite în colecţiile existente

Cele trei formate identificate în resursele partenerilor sunt: formatul PHS/LAB, formatul MULTEXT/TEI şi formatul TEXTGRID.

2.1 Formatul 1 (PHS/LAB)

Formatul PHS/LAB este reprezentat de calupuri de câte 4 fişiere cu următoarele caracteristici:

- Un fişier *.wav* ce conţine înregistrarea audio a intrării;
- Un fişier *.txt* ce conţine transcrierea textului, rostit în cadrul înregistrării audio;
- Un fişier *.lab* ce conţine varianta procesată a textului din fişierul *.txt*, în care sunt eliminate semnele de punctuaţie;
- Un fişier *.phs* care conţine o aliniere la nivel de fonem a textului transcris. Pentru fiecare fonem este precizat momentul începerii rostirii lui, respectiv momentul terminării rostirii.

Fişierul *.phs* este structurat pe 4 coloane, astfel: (1) prima coloană reprezintă momentul începerii pronunţării fonemului; (2) a doua coloană prezintă momentul terminării rostirii fonemului; (3) pe a treia coloană se regăseşte fonemul; (4) a patra coloană este prezentă doar în cazul fonemelor ce marchează începutul cuvintelor şi cuprinde cuvântul întreg, ce urmează a fi rostit.

```
0 3700000 pau pau
3700000 4400000 p purt\304\203torul
4400000 4900000 u
```

4900000 5200000 r
5200000 5600000 t
5600000 6200000 @
6200000 6700000 t
6700000 7200000 o
7200000 7500000 r
7500000 8100000 u
8100000 8400000 l
8400000 8800000 d de
8800000 9200000 e

Textul de mai sus reprezintă un extras dintr-un fişier *.phs*. De remarcat este modul în care se salvează diacriticele. De exemplu, litera "ă" este reprezentată printr-o serie de caractere "\304\203". Acest lucru poate genera o eroare la parsarea acestor fişiere, dacă nu sunt tratate corespunzător diacriticele.

Un dezavantaj al acestui format este faptul că nu prezintă metadate legate de sursa înregistrării sau detalii tehnice despre fişierul audio. Astfel, pentru a converti fişierele din acest format în codificarea agreată în raportul A1.3 [3], este necesară adăugarea acestor informaţii.

Fişierele .wav prezintă în general înregistrării la nivel de propoziţie. Acest lucru este util, deoarece permite stocarea unei alinieri atât la nivel de propoziţie, cât şi la nivel de fonem, folosindu-se informaţiile din fişierul *.phs*.

3.2.1 Formatul 2 (MULTEXT/TEI)

MULTEXT (*Multilingual Text Tools and Corpora*) este un proiect de cercetare finalizat ce şi-a propus să sprijine dezvoltarea şi utilizarea unor resurse multilingve prin proiectarea de standarde de adnotare, dezvoltarea de tehnologii de prelucrare şi crearea unor corpusuri. Acest proiect a fost continuat prin MULTEXT-EAST [1] ce a avut aceleaşi obiective, de data asta însă pentru limbile vorbite în estul Europei. Una din resursele create în acest proiect este şi un corpus aliniat voce-text ce va fi încărcat pe platforma CoBiLiRO. În acest context, acest format a fost analizat şi a fost investigată o modalitate de armonizare a sa cu formatul agreat în raportul A1.3 [3].

În vederea adnotării corpusului aliniat, MULTEXT-EAST a plecat de la standardul propus de TEI 4.0 [2] pentru textul propriu-zis, la care a adăugat un marcaj specific pentru înregistrarea audio corespunzătoare.

```
<div id="sro.2" n="O1" type="block">
<head>*BLOCK: O1</head>
<p id="sro.2.2"><s id="sro.2.2.1">Am o problem&#x103; cu aparatul de dedurizare a apei.</s>
```

```
...</p>
<ab>[<xref url="../spc/spch01-ro.wav">speech file</xref>]</ab></div>
```

Spre deosebire de formatul CoBiLiRO, unde se preferă o aliniere unu-la-unu între fişierul text şi cel audio, în acest format textul este asociat cu mai multe fişiere .wav, corespunzătoare fiecărui paragraf. O resursă încărcată pe platformă în acest format va fi compusă dintr-un fişier *.xml* şi un set de fişiere *.wav* cu scurte înregistrări audio. Fişiere audio nu au o calitate deosebită (doar 16kbps), dar conţin o singură voce, fără alte zgomote şi au dimensiuni relativ mici (aproximativ 1MB pentru un paragraf). Header-ul TEI inclus în fişierul XML conţine majoritatea informaţiilor cerute în formatul CoBiLiRo, iar codificarea este similară. Conţinutul de text şi alinierea sunt cuprinse în elementul `<text>` ce conţine şi atributele `id` (un identificator unic pentru document) şi `lang` (limba textului din document). Textul este segmentat în paragrafe, iar pentru a marca legătura dintre un paragraf şi fişierul/fişierele audio este folosit elementul `xref`, definit în standardul TEI. Descrierea elementelor şi atributelor este bine documentată, tagset-ul folosit fiind TEI 4.0. Formatul fişierului XML, exclusiv header-ul TEI, este prezentat mai jos.

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xs:element name="TEI.2">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="text"/>
      </xs:sequence>
      <xs:attribute name="id" use="required" type="xs:NCName"/>
      <xs:attribute name="lang" use="required" type="xs:NCName"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="text">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="body"/>
      </xs:sequence>
      <xs:attribute name="id" use="required" type="xs:NCName"/>
      <xs:attribute name="lang" use="required" type="xs:NCName"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="body">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="div"/>
      </xs:sequence>
      <xs:attribute name="lang" use="required" type="xs:NCName"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="div">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="head"/>
        <xs:element ref="p"/>
        <xs:element ref="ab"/>
      </xs:sequence>
      <xs:attribute name="id" use="required" type="xs:NCName"/>
      <xs:attribute name="n" use="required" type="xs:NCName"/>
      <xs:attribute name="type" use="required" type="xs:NCName"/>
    </xs:complexType>
  </xs:element>
```

```
  <xs:element name="head" type="xs:string"/>
  <xs:element name="p">
    <xs:complexType>
      <xs:sequence>
        <xs:element maxOccurs="unbounded" ref="s"/>
      </xs:sequence>
      <xs:attribute name="id" use="required" type="xs:NCName"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="s">
    <xs:complexType mixed="true">
      <xs:attribute name="id" use="required" type="xs:NCName"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="ab">
    <xs:complexType mixed="true">
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" ref="xref"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="xref">
    <xs:complexType mixed="true">
      <xs:attribute name="url" use="required"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

3.2.2 Formatul 3 (TEXTGRID)

Cel de-al treilea format identificat, formatul TEXTGRID, este alcătuit din 3 fişiere:

- Un fişier *.wav* ce conţine înregistrarea audio a intrării;
- Un fişier de tip *.TextGrid* care conţine o aliniere a tuturor vocalelor din text. Rândurile trei şi patru din fişier precizează momentul începerii rostirii şi momentul terminării rostirii (exprimate prin două variabile globale - *xmin* şi *xmax*, respectiv timpul de început şi sfârşit). Ulterior, este marcat un vector de elemente (*items*), care sunt vocalele din text, fiecare având un interval - cu xmin, xmax şi vocala transcrisă fonetic;
- Al treilea fişier este de tip *.txt* şi conţine locaţia fişierului, dimensiunea sa în kilobiţi, data înregistrării, precum şi un vector de valori care prezintă, pentru fiecare vocală identificată:
    a. un identificator
    b. durata în milisecunde
    c. energia măsurată în decibeli
    d. frecvenţa pentru formatul *F1*, *F2* şi *F3* măsurată în Hz

Exemplu format TextGrid

```
File type = "ooTextFile" - tipul fişierului
Object class = "TextGrid" - formatul fisierului

xmin = 0
```

```
xmax = 1.8284375
tiers? <exists>
size = 1
item []:
    item [1]: - indicele colecţiei
        class = "IntervalTier"
        name = "voyelles"
        xmin = 0
        xmax = 1.8284375
        intervals: size = 17
        intervals [1]:
            xmin = 0
            xmax = 0.21698313449079956
            text = ""
        intervals [2]:
            xmin = 0.21698313449079956
            xmax = 0.3118133965778926
            text = "e"
        intervals [3]:
            xmin = 0.3118133965778926
            xmax = 0.3553420414703287
            text = ""
        intervals [4]:
            xmin = 0.3553420414703287
            xmax = 0.5139106764356318
            text = "\'1a:"
        intervals [5]:
            xmin = 0.5139106764356318
            xmax = 0.590085804997395
            text = ""
        intervals [6]:
            xmin = 0.590085804997395
            xmax = 0.6926890393867089
            text = "a:"
        intervals [7]:
            xmin = 0.6926890393867089
            xmax = 0.7517636288835865
            text = ""
        intervals [8]:
            xmin = 0.7517636288835865
            xmax = 0.8792403746400065
            text = "\'1\ef:"
        intervals [9]:
            xmin = 0.8792403746400065
            xmax = 0.916550641690666
            text = ""
        intervals [10]:
            xmin = 0.916550641690666
            xmax = 1.0160446871590916
            text = "\ic"
        intervals [11]:
            xmin = 1.0160446871590916
            xmax = 1.095329004641743
            text = ""
        intervals [12]:
            xmin = 1.095329004641743
            xmax = 1.2787711509741526
```

```
            text = "\'1a::"
        intervals [13]:
            xmin = 1.2787711509741526
            xmax = 1.3689376296799132
            text = ""
        intervals [14]:
            xmin = 1.3689376296799132
            xmax = 1.473095458529671
            text = "\sw:"
        intervals [15]:
            xmin = 1.473095458529671
            xmax = 1.5197332923429954
            text = ""
        intervals [16]:
            xmin = 1.5197332923429954
            xmax = 1.6254457156531976
            text = "i\nva:"
        intervals [17]:
            xmin = 1.6254457156531976
            xmax = 1.8284375
            text = ""
```

```
D:\Arad\9F75twpi3.txt  size: 29255    - locaţia fişierului
28-Aug-2013    - data înregistrării

                       duration [ms]          energy [dB]            fo1     fo2     fo3 [Hz]
1                             95                    62              203     202     200
2                             159                   62              199     193     191
3                             103                   64              191     183     182
4                             127                   62              180     171     167
5                             99                    61              173     161     162
6                             183                   55              168     168     185
7                             104                   62              295     305     297
8                             106                   58              262     259     259

values at:
3472 4230 4989 5685 6954 8223 9441 10262 11083 12028 13048 14068 14665 15461 16257 17525 18993
20460 21903 22736 23570 24316 25161 26007
```

Pentru fiecare valoare de tip *interval,* se va genera tag-ul *unit* aferent, completat cu atributele *start* şi *stop* luate din câmpurile *xmin* şi *xmax.* Atributul *text* va fi mapat folosind câmpul *text* din fişierul *textgrid*.

Fiecare înregistrare de tipul :

```
intervals [1]:
            xmin = 0
            xmax = 0.21698313449079956
            text = ""
```

va fi convertită la o înregistrare specifică *teiHeader*:

```
<teiHeader>
      <unit>
         <speech start=”0.0000” stop = ”0.21698313449079956”/>
         <text></text>
      </unit>
</teiHeader>
```

3.3 Soluţii de armonizare propuse

3.3.1 Formatul 1 (PHS/LAB)

Având în vedere că *formatul PHS/LAB* nu conţine niciun fel de metadate dintre cele cerute de header-ul, inspirat de standardul TEI, al formatului CoBiLiRo, partenerul care va încărca resursa pe platformă va trebui să completeze în interfaţa grafică câmpurile aferente acestor metadate. Valorile lor vor fi preluate şi stocate în header.

Textul aferent fiecărei înregistrări audio va fi extras din fişierul *.txt* şi va fi convertit pentru formatul *xml* CoBiLiRo într-un tag de tip *unit*. Acest tag va conţine şi numele fişierului audio aferent textului. De asemenea alinierea se poate face şi la nivel de fonem, folosind informaţia din fişierul *.phs*, unde există două valori, ce indică începutul şi sfârşitul unui fonem în milisecunde.

Pe lângă fişierul *xml* ce conţine metadatele şi informaţia textuală, platforma va stoca separat şi fişierele audio. În tabelul de mai jos prezentăm o posibilă mapare a elementelor formatului PHS/LAB la elementele formatului CoBiLiRo.

Format PHS/LAB	**Format CoLiBiRo**	
Valoare	***tag***	***atribut***
Numele fişierului audio	`Unit>speech`	`speechFile`
Conţinutul fişierului *.txt*	`Unit>text`	
Prima coloana a fişierului *.phs*	`Unit>speech`	`start`
A doua coloana a fişierului *.phs*	`Unit>speech`	`stop`
A treia coloana a fişierului *.phs*	`Unit>text`	

3.3.2 Formatul 2 (MULTEXT/TEI)

Spre deosebire de primul format, formatul MULTEXT prezintă o serie de metadate, descrise într-o manieră similară cu cele din formatul agreat în raportul 1.3 [3]. Astfel, pentru armonizarea resurselor disponibile în acest format, primul pas ar fi selectarea din xml-ul aferent formatului 2 a acelor câmpuri de care ne putem folosi pentru a completa header-ul formatului CoBiLiRo. Aceste date vor fi preluate automat, rămânând ca utilizatorul ce încarcă resursa să completeze în interfaţa platformei şi celelalte câmpuri necesare, care nu sunt incluse în formatul MULTEXT.

Conţinuturile fişierelor audio vor fi preluate din xml-ul formatului 2 (din tag-ul *text* ce prezintă o serie de tag-uri de tip *div* pentru textul fiecărui fişier audio) şi introduse într-un tag de tip *unit* alături de denumirea fişierului audio aferent textului. Fişierele audio vor fi reţinute şi ele, separat de fişierul *xml* TEI.

Format 2	**Format 1.3**	
Valoare	***tag***	***atribut***
`text>div>ab`	`Unit>speech`	`speechFile`
`text>div>p`	`Unit>text`	
`teiHeader>fileDesc>titleStmt>title`	`teiHeader>title`	

3.3.3 Formatul TEXTGRID

Acest format este folosit de aplicaţia de analiză şi adnotare a semnalului audio Praat [4], una din cele mai populare în domeniu, cu o istorie de aproape 20 de ani. La fel ca Formatul 1 (*PHS/LAB*), formatul TextGrid nu conţine niciun fel de metadate conform cu cele prezente în header-ul standard.

Aceste metadate vor fi completate de către utilizator, folosind interfaţa grafică. Datele din fişierele *txt* şi *textgrid* vor fi salvate în tabele specifice.

3.4 Soluţii tehnice pentru conversia automată

După identificarea formatelor iniţiale şi a celui final, precum şi a soluţiilor de armonizarea avute în vedere, propunem câteva soluţii tehnice pentru realizarea unei conversii în care intervenţia utilizatorului să fie minimă.

Anterior etapei de conversie are loc o etapă de validare în care o serie de verificări sunt făcute automat peste toate câmpurile resursei, întâi pe *client* şi apoi pe *server*.

Ulterior sunt validate şi fişierelor asociate unei resurse. Altfel, platforma trebuie să se asigure în primă fază că fişierele urcate respectă măcar unul dintre formatele cunoscute. Dacă se

constată prezenţa unui format cunoscut se trece mai departe la validarea corectitudinii formatului. Asta înseamnă parcurgerea fişierelor de intrare, verificarea conţinutului lor şi atenţionarea contributorului asupra problemelor apărute. Validările se preocupă de aspecte precum: tipul fişierelor audio (acestea ar putea avea un format neagreat), numele şi extensiile fişierelor, lipsa unor informaţii din fişierele urcate (lipsa unor tag-uri sau atribute XML etc.) etc. Aceste validări s-ar putea face atât la nivelul clientului (în browser), ceea ce ar însemna un consum în plus de memorie a maşinii ce găzduieşte fişierele, sau la nivelul serverului, ceea ce implică trimiterea fişierelor către server, procesarea lor şi trimiterea înapoi la client a unei eventuale liste de erori.

O altă validare foarte importantă ce ar trebui făcută la nivel de client, înainte de trimiterea fişierelor către server, este aceea a verificării extensiilor fişierelor urcate. Un utilizator ar putea încărca un fişier maliţios, care odată ajuns pe server, să îi afecteze integritatea sau chiar să-l compromită. De aceea ar trebui să avem o listă predefinită de extensii acceptate pe server.

Un utilizator poate alege să încarce un director de pe disc cu fişierele sale sau o arhivă care să le conţină pe toate. Pentru acest ultim caz, platforma va fi capabilă să dezarhiveze fişierele şi să le treacă prin procesul de validare şi mai apoi upload. Această validare ar trebui făcută la nivel de client deoarece urcarea lor pe server şi dezarhivarea lor acolo ar însemna încărcarea unor fişiere maliţioase ce s-ar putea găsi în arhivă.

În cazul încărcării unor resurse de mari dimensiuni, tot acest proces ar putea dura foarte mult. Astfel, un contributor nu trebuie neapărat să urmărească evoluţia procesului din faţa calculatorului. Poate să pornească procesul şi să revină oricând pe pagina platformei pentru a verifica starea procesărilor. De asemena, la sfârşit, va primi o notificarea ce îl va informa dacă procesul s-a realizat cu succes sau dacă au apărut erori la validare.

Deoarece vorbim de resurse audio, care pot fi semnificative ca dimensiune, ne punem întrebarea ce probleme ar putea apărea dacă dimensiunea fişierelor depăşeşte un 1 GB. Într-un asemenea caz majoritatea serverelor s-ar opri datorită unui time-out mult prea mare sau memoria serverului s-ar putea încărca peste o anumită limită admisă. Posibile soluţii ar fi modificarea acestor limite sau spargerea input-ului în arhive mici, urcarea lor pe rând pe server şi reasamblarea lor în backend.

Serverul pe care va fi instalată aplicaţia platformei este modelul HP ProLiant ML350 Gen9, cu un procesor E5-2650 v3 de 2.30GHz, care conţine 40 de core-uri.

Având acest număr de core-uri, aplicaţiile se pot împărţi pe fire de execuţie (thread-uri). O aplicaţie nu consumă mai mult de 5 fire de execuţie la o rulare. Având la dispoziţie 40 de core-uri, nu vor fi ocupate simultan toate, întrucât 15 core-uri vor fi necesare pentru rularea sistemului de operare şi a altor programe necesare acestuia şi se recomandă să existe un minim de 5 core-uri care să fie lăsate libere pentru orice altă procesare care poate fi realizată de către server - update, generare de log-uri etc. Astfel, avem la dispoziţie un număr de 20 de core-uri pentru aplicaţie, adică 4 procese a câte 5 fire de execuţie.

.NET Core, *jQuery* şi *MariaDB*

Principalul motiv pentru care s-a ales framework-ul ASP.NET Core a fost faptul că tehnologia este *open-source, necesită resurse de calcul rezonabile, este rapidă şi modulară*. Acest framework este în plină dezvoltare şi ne pune la dispoziţie o serie de librării şi pachete *NuGet* ce ne ajută la dezvoltarea rapidă a platformei.

Unul din beneficiile majore aduse de .NET Core este portabilitatea. Acest lucru ne permite să hostăm aplicaţia noastră pe orice sistem de operare.

Un alt motiv pentru care am folosit această tehnologie a fost faptul că ne oferă o securitate a datelor (protecţie pentru atacuri de tip *SQL Injection* şi *Cross-site request forgery)* şi un mod de a securiza API-urile REST folosind *JSON Web Token.*

Performanţa ne este garantată de faptul ca aplicaţia poate fi scalată pe anumite servicii (de exemplu, putem aloca un număr mai mare de instanţe pentru serviciile de conversie - care vor fi mai costisitoare din punct de vedere al procesării).

Pentru programarea la nivel de client - *client-side* s-a folosit framework-ul jQuery. Acesta ne permite manipularea DOM-ului şi accesul la anumite animaţii şi validări specifice.

3.5 Concluzii

În acest raport au fost descrise trei formate identificate ca fiind disponibile pentru a fi încărcate de parteneri pe platforma CoBiLiRo şi au fost propuse soluţii de armonizare în scopul oferirii de către platformă a unui format standardizat, ce facilitează valorificarea întregii colecţii de resurse. Formatele originale nu au satisfăcut necesităţile platformei, aşa cum au fost descrise în raportul corespunzător activităţii A1.3 [3]. Formatul propus în acel raport serveşte ca standard de adnotare la care vor fi convertite celelalte formate, începând cu cele trei descrise în acest raport. În cazul acestor conversii este necesară şi completarea informaţiilor disponibile în fişierele originale, în special în zona de header, unde informaţii precum autorul, sursa textului şi a înregistrării audio sunt esenţiale pentru o platformă de distribuire precum cea dezvoltată în cadrul proiectului. În cazul identificării ulterioare şi a altor formate relevante, soluţii similare de armonizare pot fi propuse. De asemenea, în cazul în care vor fi necesare ajustări ale standardului propus pentru platforma CoBiLiRo, soluţiile automate de conversie vor fi adaptate.

Toate obiectivele incluse în plan la această activitate au fost realizate.

Bibliografie

[1] Dimitrova, L., Ide, N., Petkevic, V., Erjavec, T., Kaalep, H. J., & Tufis, D. (1998, August). Multext-east: Parallel and comparable corpora and lexicons for six central and eastern european languages. In *Proceedings of the 36th Annual Meeting of the Association for Computational*

Linguistics and 17th International Conference on Computational Linguistics-Volume 1 (pp. 315-319). Association for Computational Linguistics.

[2] Ide, N., & Véronis, J. (Eds.). (1995). *Text encoding initiative: Background and contexts* (Vol. 29). Springer Science & Business Media.

[3] Cristea, D., Scutelnicu A. (2018, noiembrie), Raport *"Activitatea A1.3: Proiectarea funcţională şi arhitecturală a infrastructurii care va găzdui resursele şi instrumentele de prelucrare şi acces ale consorţiului şi realizarea unui prototip"*, proiect RETEROM

[4] Boersma, P. (2002). Praat, a system for doing phonetics by computer. Glot international, 5.