

TWINNING CONTRACT

BA 12 IB ST 01

Support to the State and Entity Statistical Institutions, phase VI



MISSION REPORT

on

Activity 2.1.5 Enterprise Groups II

Mission carried out by
Søren Netterstrøm, Statistics Denmark
Søren Kristensen, Statistics Denmark

17- 21 October 2016

Version: Final

Expert contact information

*Søren Netterstrøm
Statistics Denmark
Farverstræde 6
DK-4850 Stubbekøbing
Denmark
Phone: +45 54 44 04 75
Email: SNE@VIV.dk*

*Søren Kristensen
Statistics Denmark
Sejrøgade 11
DK-2100 Copenhagen Ø
Denmark
Phone: +45 39 17 33 37
Email: SKR@dst.dk*

Table of contents

1. General comments.....	3
2. Assessment and results.....	4
3. Conclusions and recommendations.....	7
Annex 1. Terms of Reference	8
Annex 2. Persons met.....	9
Annex 3. Data model (conceptual).....	10
Annex 4. Creating EGs from relations	12
Annex 5. Algorithms for updating EGs, first draft.....	18
Annex 6. Pseudo Language.....	24

List of Abbreviations

BHAS	Agency for Statistics of Bosnia and Herzegovina
BiH	Bosnia and Herzegovina
CBBH	Central Bank of Bosnia and Herzegovina
EC	European Commission
EU	European Union
FBiH	Federation of Bosnia and Herzegovina
FIS	Institute for Statistics of Federation of Bosnia and Herzegovina
MS	EU Member State
RSIS	Institute for Statistics of Republika Srpska
RTA	Resident Twinning Adviser
TOR	Terms of Reference

1. General comments

This mission report was prepared within the Twinning Project “Support to the State and Entity Statistical Institutions, Phase VI”.

The main purpose of the mission was to assist the BiH State and Entity Statistical Institutions and the District of Brčko in their efforts to introduce enterprise groups in the SBR. It is the consultants’ point of view, that the EG structure should be created and introduced in the SBR on the basis of a data set that encompasses BiH as a whole. This solution does not preclude the creation of truncated enterprise groups (TEG) at Entity level if this is considered necessary. It is as of yet not clear which data sources are available and whether they contain the necessary information, but the work on the mission has been carried out on the assumption that it is possible to obtain data which is adequate for the purpose.

The consultants would like to express their thanks to all officials and individuals they have met for the kind support and valuable information which they have received during their stay in BiH. They would also like to thank for the very active participation in discussions which revealed both a high level of engagement and broad knowledge of the area-

The views and observations stated in this report are those of the consultants and don’t necessarily correspond to the views of EU, BHAS/FIS/RSIS or Statistics Denmark.

2. Assessment and results

2.1 Methodology

BHAS has prepared a first version of a methodology paper that was presented and discussed. The paper in a very good way discusses the basic concepts of Enterprise Groups and includes some very illustrative examples. It also contains a list of the variables that should (optionally) be included for the different unit types. This was very useful in discussing the data model (see later).

The paper should of course be updated with a description of the operational level.

2.2 Input for Enterprise Groups

The sources for the information needed to create Enterprise Groups have not been identified at this stage. The Methodology paper names some potential sources, others were briefly discussed during the meeting. This however requires much more knowledge on the availability, content and coverage of the potential sources than what is presently available. There may also be different sources for each of the Entities and the District of Brčko.

Whatever the sources are, the goal is to create a file giving relations between legal units and a file with basic information for Foreign Legal Units.

The relation should contain ID of Parent and Child and share of Ownership. The ID numbers should be Legal Unit Number (for units within BiH) or ID Number used in a special table for Foreign Legal Units, that needs to be created in order to cover Multinational Enterprise Groups.

Once sources have been identified the procedures to create the above mentioned files must be established.

2.3 Creation Enterprise Groups from the relations

The consultants demonstrated how it is possible from the input mentioned above to create Enterprise Groups including both direct and indirect control based on the method used in Denmark.

Appendix 4 contains more detailed information on this topic.

2.4 Data model

BHAS had prepared a first draft of a data model based on the discussion in the previous mission. Together with the list of variables in the methodology paper this was used to draw an updated conceptual data model. The model must be transformed into a 'real' data model, but at this stage the conceptual model gives a more useful picture of what to achieve.

The model includes 5 objects. The Enterprise Group, (resident) Legal Unit, Foreign Legal Unit, EnterpriseGroup-Legal Unit relation and LegalUnit-LegalUnit relation. Whereas the Legal unit object already exists in the SBR, the others are new units which must be created for the purpose of including Enterprise Groups in the SBR.

Variables were assigned to the different units and it was demonstrated how this would meet the requirement in the Methodology.

Three variables give special problems; activity code, turnover and employment.

- In order to create the ActivityCode (2 digit NACE) of the EG all legal units and their turnover needs to be considered. It may be possible to do on the basis of Turnover-classes but this is not optimal.
- According to the regulation total employment and turnover for the group is also needed but this requires that this information (on level of truncated enterprise group for each entity (that may be a single legal unit/enterprise) is exchanged in the SBR system. If a solution for this cannot be found these variables cannot be created.

The consultants suggested that it may be possible to take an approach similar to the method that was proposed and agreed for exchanging data on employment from the Pensions Funds , where data was to be received based on the pension fund of the employed person and had to be exchanged in cases where the employer (legal unit) was in another entity (District of Brčko

In this case, once the Enterprise Groups have been created, the responsibility for an enterprise group should be assigned to one of the entities based on a rule. For all legal units belonging in another entity data on turnover and employment is passed on to the responsible entity.

The consultants recommend to further examine this option.

Appendix 3 will give more detailed information.

2.5 Updating Enterprise Group Register (SBR)

It is assumed that information about relations of legal units will be collected on a yearly basis. Based on this information the Foreign Legal Units are updated and a complete set of data for the current year about Enterprise Groups are created as described above.

Once this data are produced they are used to update the Enterprise Group Register. This involved the creation of New Enterprise Group, Updating Enterprise Groups (adding and/or deleting Legal Units from the group and updating information on turnover/employment) and terminating Enterprise Groups. The register is designed to hold current and historical information, i.e. the option to see how an Enterprise Group evolved over time.

There will be a need to describe and develop software (procedures) to undertake this task. The consultants suggested to use the methods used when the SBR was developed, that is to use 'pseudo-language' to describe the processes before they are actually implemented (most likely in T-SQL). Several members of the working group have been part of the SBR development and know the method. A short introduction was given by the consultant.

Within the timeframe of the mission it was not possible to go in much detail with how this update should be performed, just some simple examples could be given.

In most cases it is assumed that the changes are 'simple' and can be handled automatically, i.e. the creation of a new group, the addition or deletion of a few members or termination. However, more complex cases like exchange of Legal units between several Enterprise Groups may not be possible to describe in advance how to handle. The method of 'unfinished' transactions should be used to refer such cases for manual inspection and handling.

Creating the full set of rules may be time consuming and the consultants propose that from the offset 'complex' cases are not further developed. When the system then has to be updated it may be necessary to fine tune the system to capture more cases for automatic processing.

2.6 Online system

Once the register is established it should be possible to examine units in the system from an online system. The consultants propose to limit the system to browsing as manual updates of enterprise groups is expected to be very rare and thus could be handled by 'ad-hoc' solutions gradually building a set of procedures for the most common situations.

3. Conclusions and recommendations

A lot of work has been done at the conceptual level, but some steps are needed in order to move to a more operational level.

The most immediate task is to investigate and analyse potential data sources for the creation of enterprise groups. The preferred option is to get the data from one source, but if better information can be obtained by combining sources, then that is obviously also an option. Not having seen the data, the consultants are not in a position to give any advice on how to prioritise sources.

If it is considered feasible to obtain input data, work on the algorithm for the creation and updating of enterprise groups can be started even if adequate input data is not yet available. It is recommended that this work is carried out on test data (or a small subset of real input data). This is often a more efficient way of developing and testing an algorithm. The work should take departure in the simple situations and gradually consider what should be done in case of more complex events. It is recommended that the aim should not be to create a perfect algorithm that can handle all possible complex situations and business demographic events, but rather to adopt an incremental approach and expand the algorithm gradually.

A third step is to decide on the data model and start implementing this in the SBR. This can also be commenced before the input data is available.

Annex 1. Terms of Reference

EU Twinning Project BA-12-IB-ST-01

Terms of Reference

Component: 2.1. Statistical Business Register

17 – 21.10.2016.

Institute for Statistics of Republika Srpska, Banja Luka

Activity 2.1.5

Enterprise Groups I

Benchmarks

- 5-year development plan updated by 1st project quarter and by 8th project quarter
- Methodology for linking SBR with the compilation of SBS business demography characteristics available by 4th project quarter
- **Methodology for enterprise groups in SBR available by 8th project quarter**
- Institutional sectors code assigned to the SBR units by 8th project quarter
- SBR is ready to be used for analysis of business demography by 8th project quarter

Purpose of activity

The expected activities are:

- Introductory speech on Enterprise Groups preparing by experts
- Presentation and discussion on draft methodology for Enterprise Groups
- Discussion of the present status (sources, data etc.) on Enterprise Groups in Bosnia and Herzegovina
- Discussion of technical preconditions and solutions on introductions of EG in SBR data model
- Next mission
 - Preparation of the list of activities to be done before the next mission

Expected output

- General overview on the present status on Enterprise Groups in Bosnia and Herzegovina
- Methodological and practical questions on introduction of EG in BiH which are requested by participants, discussed and explained by MS Experts
- Input to final methodology made
- Technical preconditions and solutions on introductions of Enterprise Groups in SBR assessed
- The list of activities to be done before the next mission, prepared and agreed by all partners (BHAS-BD/ FIS/RSIS participants, experts and RTA)
- Draft ToR for the next mission prepared and agreed by all partners (BHAS-BD/ FIS/RSIS participants, experts and RTA)

Annex 2. Persons met

Agency for Statistics of BiH

Statistical Business Register:

Dženita Mustafić	Specialist for SBR (Coordinator for SBR Component, BHAS)
Mevlija Odošić	Senior Advisor for SBR
Ivana Tavra Čolo	Senior Official for SBR
Senija Fačić	Head of Branch Office in Brčko District
Vedad Osmanović	Senior Advisor for IT in Brčko District

Institute for Statistics of Federation of BiH (FIS)

Statistical Business Register:

Enisa Rastić	Head of Unit for SBR (Coordinator for SBR Component, FIS)
Razija Bičakčić	Senior Advisor for IT

Institute for Statistics of Republika Srpska (RSIS)

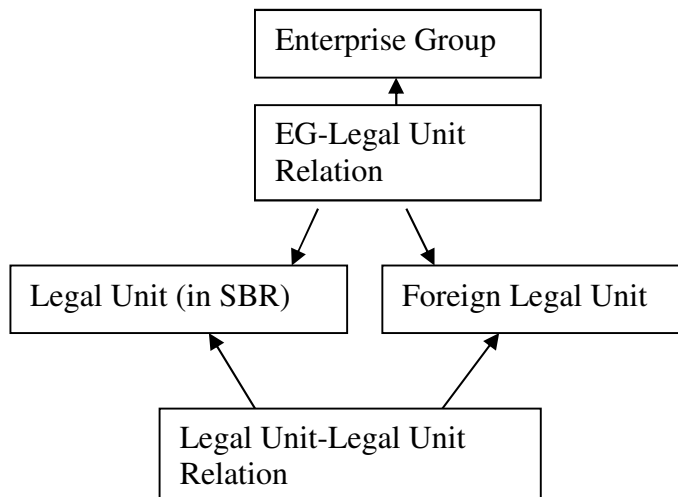
Statistical Business Register:

Nataša Teinović	Senior Official for SBR
Mladen Marić	Head of data entry, application development and database administration

Twinning Project Administration

Søren Leth-Sørensen	RTA
Djemka Šahinpašić	RTA Assistant
Haris Imamović	Interpreter

Annex 3. Data model (conceptual)



The five basic units needed to create Enterprise Groups.

The Enterprise Group should have the following variables

- IDNumber (assigned by BiH)
- Idnumber and from Eurogroup (optional)
- Name in BiH
- Type of group (Multinational, Domestic)
- Dates of commencement/cessation
- 2 digit Nace code
- Employees/turnover in BiH

The Enterprise Group-Legal Unit Relation should have the following variables

- IDNumber of Enterprise
- Type of Legal Unit (Domestic/Foreign)
- ID of Legal Unit
- Role of Legal Unit
 - Domestic head
 - Domestic other member
 - Foreign Head and First
 - Foreign Head
 - Foreign First
 - Foreign other member (controlled by BiH unit)
- Dates of commencement/cessation of relation (1.9/1.10)

The Legal Unit in SBR has no new variables assigned

The Foreign Legal Unit should have the following variables

- IDNumber
 - If there is a register within TAX their ID number may be used
 - or ID number may be assigned by Statistics
- Name
- Address (req if first legal unit)
- Country
- ID number in Country (SBR)
- TAX/VAT number (in Country)

The Legal Unit-Legal Unit relation should have the following variables

- Parent_Unit_type (Domestic/Foreign)
- Parent_Unit ID
- Child_Unit_type (Domestic/Foreign)
- Child_Unit_ID
- Dates of creation/cessation
- Share of Ownership

Note. The list of attributes reflects the minimum needed to fulfill the requirement of the EU manual. Others may be added as needed, i.e. the responsible Entity on Enterprise Group.

When creating the database design, UID should be used and links expressed and fk_.. (similar to SBR), where the conceptual model uses the ID number.

Annex 4. Creating EGs from relations

Input to the process is a table (LEG_REL) that contains

Parent_ID

Child_ID

Ownership%

Example

Parent	Child	Percent
1	2	60
1	3	60
2	4	90
4	5	80
3	6	30
4	6	30
7	6	40

STEP 1

From this table a new table(LEG_POP) with all ID-numbers are created

At this stage Controlled and root are blank

ID number	Controlled	Root
1		1
2	1	1
3	1	1
4	1	2
5	1	4
6		6
7		7

For each row in LEG_POP it is checked if there is a row in LEG_REL where Child =ID Number and Percent > 50.

For these the rows, Controlled is changed to 1 as in the example above and Parent is set as root

For other rows IDNumber (1,6,7) root is equal IDNumber

STEP 2

Next step will examine and update root in LEG_POP.

For each row

If root <> idnumber

 Old_root = root

 New_root = select root from Leg_POP where IDNumber=old_root

 If new_root <> old_root

 Set root=new_root

 End if

End if

This step is repeated until no rows change root

Example

4 has 2 as root

2 has 1 as root next iteration (4 gets 1)

1 has 1 so no change

Now LEG_POP should look like

ID number	Controlled	Root
1		1
2	1	1
3	1	1
4	1	1
5	1	1
6		6
7		7

STEP 3

Now it is time to look for indirect control.

To do this we create a new relation table (Root_LEG)

From LEG_REL all rows where control of child (in LEG_POPI is blank

Root is taken from Leg_POP, Child and percent from LEG_REL

Select A.Root as Parent, ,B,Child, b.Percent

from LEG_POP A Join LEG_REL B on A.IDNUMBER = B.Parent

where A.Control is NULL

This looks like

Parent	Child	Percent
1	6	30
1	6	30
7	6	40

Now summarize over Parent and Child to get

Parent	Child	Percent
1	6	60
7	6	40

For each row in LEG_POP where control is blank it is checked if there is a row in Root_LEG where Child =ID Number and Percent > 50.

For these the rows, Controlled is changed to 2 and Parent is set as root

ID number	Controlled	Root
1		1
2	1	1
3	1	1
4	1	1
5	1	1
6	2	1
7		7

Now repeat step 2 to find real roots (nit need in this example)

STEP 3 (and Step2)is repeated until there are no more changes mad

Output from Step 3 becomes

This looks like

Parent	Child	Percent
7	6	40

Summarized we get the same table.

This make not lead to more changes in this case

```

SET NOCOUNT ON
--
-- This procedure requires that the table LEG_REL exists.
--
-- STEP 1
-- From this table a new table(LEG_POP) with all ID-numbers are created
--
IF OBJECT_ID ('dbo.LEG_POP', 'U') IS NOT NULL
    DROP TABLE dbo.LEG_POP

select distinct parentID as ID into parents from dbo.LEG_REL
select distinct childID as ID into children from dbo.LEG_REL

select ID, null as Controlled, null as Root into LEG_POP from parents
union select ID, null as Controlled, null as Root from children

DROP TABLE dbo.parents;
DROP TABLE dbo.children;
--
-- Determine Direct Control
--
--For each row in LEG_POP it is checked if there is a row in LEG_REL where Child =ID Number and Percent > 50.
--For these the rows, Controlled is changed to 1 and Parent is set as root
--For other rows IDNumber root is equal IDNumber
--
DECLARE @POPID int
DECLARE @PARENTID int
DECLARE @pct INT
DECLARE @indirect AS INT
DECLARE @ROOTCHANGED as int
DECLARE @OldRoot as int
DECLARE @NewRoot as int

DECLARE c1 CURSOR FOR
    select a.ID, b.ParentID, B.PCT from dbo.LEG_POP A left join dbo.LEG_REL B on A.ID = B.CHILDID

OPEN c1 -- open the cursor
FETCH NEXT FROM c1 INTO @POPID, @PARENTID, @pct

WHILE @@FETCH_STATUS =0
    BEGIN
        if @parentID is NULL OR @pct < 51
            update dbo.LEG_POP set Root = @POPID WHERE ID = @POPID
        else
            update dbo.LEG_POP set Root = @PARENTID, Controlled = 1 WHERE ID = @POPID

        FETCH NEXT FROM c1 INTO @POPID, @PARENTID, @pct
    END
CLOSE c1 -- close the cursor
DEALLOCATE c1 -- Deallocate the cursor

SET @indirect = 1 -- used to control when process is done (no more indirect relations are made)

WHILE @indirect = 1
    BEGIN
        SET @indirect = 0
    END
--
--STEP 2 Get the real root
--

```



```

set @ROOTCHANGED = 1    -- used to control when real root is found

While @ROOTCHANGED = 1
BEGIN
    DECLARE c2 CURSOR FOR
        select a.ID, a.root, b.root as newroot from dbo.LEG_POP A left join dbo.LEG_POP B on A.Root = B.id

    OPEN c2 -- open the cursor

    FETCH NEXT FROM c2 INTO @POPID, @oldroot, @newroot

    set @ROOTCHANGED = 0
    WHILE @@FETCH_STATUS=0
    BEGIN
        if @oldroot <> @newroot
        BEGIN
            update dbo.LEG_POP set Root = @NewRoot WHERE ID = @POPID
            set @ROOTCHANGED = 1
        END
        FETCH NEXT FROM c2 INTO @POPID, @oldroot, @newroot
    END
    CLOSE c2 -- close the cursor
    DEALLOCATE C2
END

--STEP 3
--Now it is time to look for indirect control.
--
IF OBJECT_ID ('dbo.LEG_REL_COPY', 'U') IS NOT NULL
    DROP TABLE dbo.LEG_REL_COPY;

SELECT c.Root as ParentID, A.ChildID, sum(A.Pct) as pct
    INTO LEG_REL_COPY
    FROM LEG_REL a join LEG_POP b on a.childID = b.id left join LEG_POP c on a.ParentID = c.id
    WHERE b.Controlled is null
    GROUP BY c.Root,a.ChildID

-- For each row in LEG_POP where controlled is NULL is checked if there is a row in LEG_REL_COPY where Child =ID Number
-- and Percent > 50.
-- For these the rows, Controlled is changed to 2 and Parent is set as root

DECLARE c3 CURSOR FOR
select a.ID, b.PARENTID, B.PCT from dbo.LEG_POP A join [dbo].LEG_REL_COPY B on A.ID = B.CHILDID
    where a.Controlled is null
OPEN c3
FETCH NEXT FROM c3 INTO @POPID, @PARENTID , @pct
WHILE @@FETCH_STATUS=0
BEGIN
    if @pct > 50
    BEGIN
        UPDATE dbo.LEG_POP set Root = @PARENTID, Controlled = 2 WHERE ID = @POPID
        SET @indirect=1
    END
    FETCH NEXT FROM c3 INTO @POPID, @PARENTID, @pct
END

CLOSE c3 -- close the cursor

DEALLOCATE c3 -- Deallocate the cursor

END
DROP TABLE dbo.LEG_REL_COPY;

```

Annex 5. Algorithms for updating EGs, first draft

During the mission the problems of updating Enterprise Groups was briefly discussed. The following is not an attempt to give a complete description of such an update but only an attempt to suggest a way to analyse and describe the update procedure.

Input to the process is the table of all relations in the new set and the table produced in the process described in annex 4 that create enterprise groups based on the relations. But input also includes all active relations in the database, both legal unit-legal unit and enterprise group-legal unit relations.

Based on this input a new set of tables can be created

Matching the old Legal Unit-Legal Unit relation and the new Legal Unit-Legal Unit relations could lead to the following table (LEU-LEU-transactions)

Type

- 0 New relation, exists only in new
- 1 Continuation, exist in both old and new
- 2 Termination, exists only in old

ParentID

ChildID

New Pct

Old Pct

Similar a match of LEG_POP (from annex 4) and Enterprise Group-Legal Unit will produce LEU-POP-Transactions

Type

- 0 New relation, exists only in new
- 1 Continuation, exist in both old and new
- 2 Termination, exists only in old

LEU ID

New Enterprise Group (taken from LEG_POP)

Old Enterprise Group (taken from Enterprise Group-Legal Unit or NULL)

Controlled (taken from LEG_POP)

It is clear that at the end of the update process in Legal Unit-Legal Unit the set of active relations should be equal to the relations in LEG_REL (type 0,1 in LEU-LEU-transactions) and in Enterprise Group-Legal Unit relations the set of all active relations should be equal to the unit in LEG_POP (type 0,1 in LEU_POP_transactions).

The problem is to determine when a New Enterprise Group (as created in Annex 4) really is a new Enterprise Group or the continuation of an existing group and if any existing Enterprise groups are to be terminated.

The approach taken here is to some extent similar to the approach used for updating SBR but far from identical due to the fact that whole groups has to be considered.

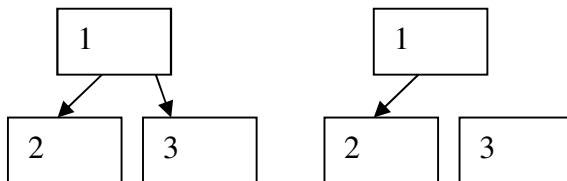
Set up simple cases

In order to analyse the task it is suggested to set up simple cases, look how they can be identified in data and finally how they should be processed.

Case 1: Legal Units that are no longer part of any Enterprise Group

Before

After



In this case number 3 is no longer part of Enterprise Group (and not part of any new group).

In this case LEU-POP-Transactions should contain

Type	LegalUnit	NewEG	OldEG	Controlled
2	3	NULL	20	NULL

It will have to be checked that 3 is not parent to any other unit.

This leads to the following

FOR EACH LEU_POP-transaction Where Type is 2

```

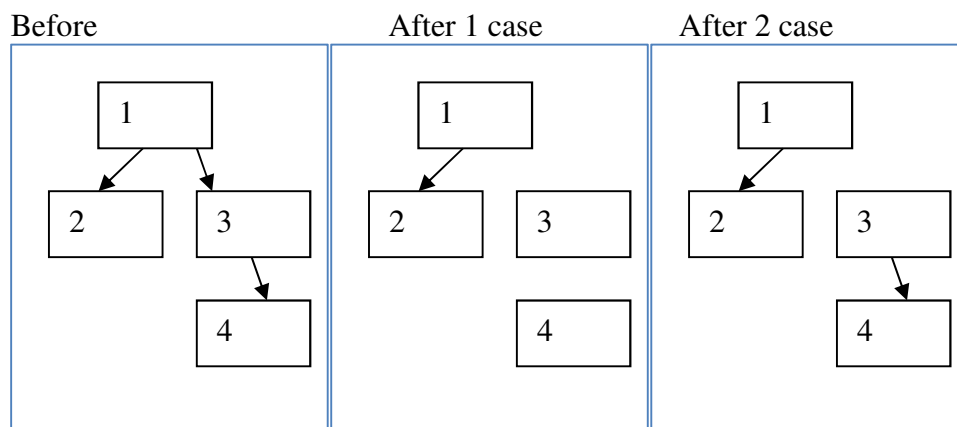
  IF there is no LEU-LEU transaction where Legal Unit is Parent
    Terminate EG-LEU relation (Cessation date=now) for LegalUnit
  IF Less than 2 relations left between EG and LEUs
    Terminate EG
  ENDIF
  Terminate LEU-LEU relation where Legal Unit is child
  Remove from LEU_POP-transaction
  Remove from LEU_LEU-transaction where Legal Unit is child
ENDIF

```

NEXT

Note, that when one or more transactions are being solved, they are removed from LEU_POP and LEU_LEU transactions.

Assume the case where 3 is parent (to 4) this could lead to two situations



The first case is solved by an iteration of the process above. In first iteration 4 would be removed, then in second iteration 3 would be removed (as the 3-4 relation is removed, thus 3 is no longer parent).

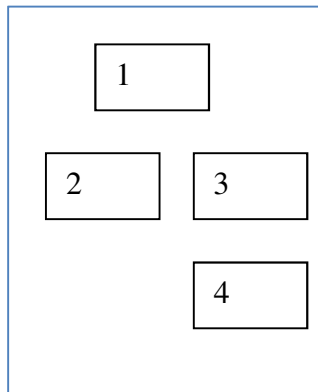
If the relation between 1 and 2 also terminates this could also be caught by the iteration above.

In the second case, there LEU_POP for legal unit 3 and 4 would be of type 1 (continuation). This is a split off and would have to be treated in another way.

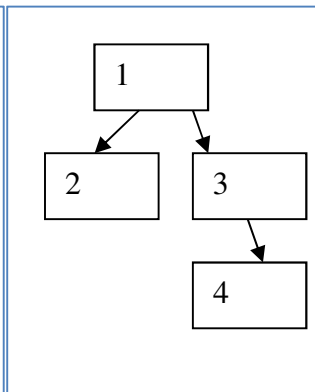
There are other possible cases. 4 could become child of 1 or part of another enterprise group or 3 could become member of another enterprise group. However, so far we are only dealing with the most simple cases.

Case 2: A completely new Enterprise group

Before



After



In this case none of the legal units before is member of any Enterprise Group but creates as new enterprise group after.

In this case LEU-POP-Transactions should contain

Type	LegalUnit	NewEG	OldEG	Controlled
0	1	10	NULL	0
0	2	10	NULL	1
0	3	10	NULL	1
0	4	10	NULL	1

To locate such cases, a new table containing ID of all new Enterprise Groups are created. Then for each group it is examined if all members is not part of any old enterprise group.

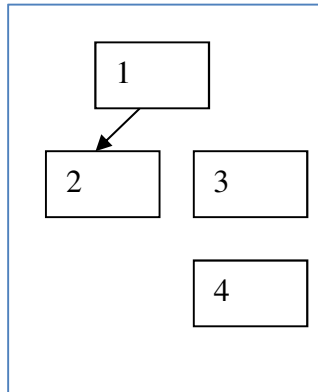
If may look like

```

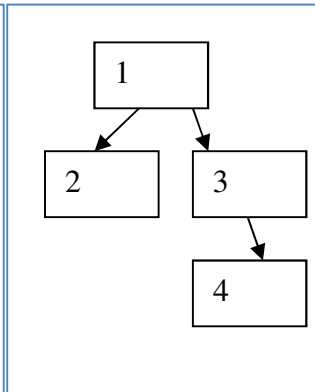
FOR EACH new Enterprise Group
  IF all LEU-POP transactions have OldEG = Null (or is type 0)
    Create New EnterpriseGroup
    FOR EACH LEU-POP transaction belonging to NEW EG
      Create LEU_POP relation
      FOR EACH LEU-LEU transaction where Legal Unit is Child
        Create LEU-LEU-relation
        Remove from LEU-LEU-transactions
      NEXT
      Remove from LEU-POP transactions
    NEXT
  ENDIF
NEXT
  
```

Case 3: New units added to enterprise group

Before



After



In this case all legal units (of new EG) before is either member of the same Enterprise Group or not member of any .

In this case LEU-POP-Transactions should contain

Type	LegalUnit	NewEG	OldEG	Controlled
1	1	10	20	0
1	2	10	20	1
0	3	10	NULL	1
0	4	10	NULL	1

To locate such cases, a new table containing ID of all new Enterprise Groups are created. Then for each group it is examined if they meet the prerequisites.

If may look like

FOR EACH new Enterprise Group

 IF all LEU-POP transactions of Type 1 has same OldEG

 And there is at least 1 type 1 and 1 type 0 and no type 2

 FOR EACH LEU-POP transaction belonging to NEW EG with type = 0

 Create LEU_POP relation

 FOR EACH LEU-LEU transaction where Legal Unit is Child

 Create LEU-LEU-relation

 Remove from LEU-LEU-transactions

 NEXT

 Remove from LEU-POP transactions

 NEXT

 FOR EACH LEU-POP transaction belonging to NEW EG with type = 1

 FOR EACH LEU-LEU transaction where Legal Unit is Child

 Update LEU-LEU-relation (if pct is different)

 Remove from LEU-LEU-transactions

 NEXT

 Remove from LEU-POP transactions

 NEXT

 ENDIF

NEXT

The examples above are just examples and may need to be further elaborated. There will be many more cases (splits, mergers, part of group moving to other group etc.). The intention of this annex is just to give an idea on how analysis and description could be done.

It is not recommended to try to capture all possible events. When all events for which the process has been predefined have been processed a number of transactions may be left unsolved. They should be examined (by enterprise groups) and handled in a semiautomatic way using in the first place ad hoc procedures. During this analysis there may be recurrent cases that can be identified. For these cases an automatic procedure may be designed and created.

Annex 6. Pseudo Language

Pseudo Language is an idea originally developed during a major revision of the Danish SBR. It has also been used in SBR projects in Croatia, BiH and Kosovo.

The main idea of Pseudo Language is to create a bridge between the subject matter specialists responsible for giving the rules for updating SBR and the IT specialist who will transform these rules into code using a formal computer language.

The basic concept is to create a basic structure to the set of rules and to enforce some strict logic without having a rigid formal language. To achieve this goal a limited set of fixed constructions are used and mixed with descriptions in 'normal' language.

The most important structure is

```
IF ... THEN
```

```
....
```

```
ELSE
```

```
....
```

```
END IF
```

An example

```
IF unit has no employment for 2 years THEN
```

```
    Mark unit as statistical dead
```

```
ELSE
```

```
    Mark unit as statistical active
```

```
END IF
```

Note that IF, THEN, ELSE and END IF are written in capital letters.

In the example above both condition and actions are written in plain language. When appropriated and it may be more formal like IF Sex = 1 THEN rather than IF Male THEN, or a by using comments

```
IF Sex = 1 THEN      * Male
```


Another structure is

```
CASE ... THEN
...
CASE ... THEN
...
CASE ... THEN
...
END CASE
```

```
Like
CASE Unmarried THEN
..
CASE Married THEN
...
CASE Divorced THEN
..
CASE Widowed THEN
.
END CASE
```

Comments or notes are an important part of Pseudo Language and should be used. There are basically three types of text, Conditions (written between IF/CASE and THEN, actions (written after THEN) and then explanatory text. The latter is important as Pseudo Language is also intended to be part of the documentation. By putting in notes and comments may be a good way to make the document easier to read and understand after a while for yourself or other persons.

Another important word is STOP.

It is assumed that rules are described 1 rule at a time. When using STOP this marks that the processing of the case handled by this rule has been done. It may be more readable than nested IF's.

So rather than

```
IF .... THEN
```

```
.....
ELSE
```

```
  IF ... THEN
```

```
  .....
  ELSE
```

```
    IF ... THEN
```

```
    ..
    ELSE
```

```
    ..
    END IF
```

```
  END IF
```

```
END IF
```

This may be easier to write and read

```
IF .... THEN
```

```
  ....
  STOP
END IF
```

```
IF .... THEN
```

```
  ....
  STOP
END IF
```

```
IF .... THEN
```

```
  ....
  STOP
END IF
```

```
.....
```

There may be other constructs like LOOP END/ LOOP or FOR EACH / NEXT or whatever is considered appropriate to fully describe a procedure.