An Implementation of a Classification Algorithm for Houses

drs. C.W. Quak

GISt Report No. 6
Delft, March 2002

Summary: *This report gives a technical description of the project 'Woningindex'. It is supplementary to the Dutch Cadastre Report 'Woningindex Haalbaarheidsstudie', which is about the organizational and feasability aspects of the automatic classification of houses. This report focusses on the technical and implementation aspects of the study, and includes a description of input data, parameters, scripts and dataflow of the prototype implementation that was built at the Delft University of Technology.*

Faculty of Civil Engineering and Geosciences
Department of Geodesy, Section GIS Technology
Thijsseweg 11, 2629 JA Delft, the Netherlands
http://www.gdmc.nl/

Contents

List of Figures

List of Tables

1 Introduction

This documents gives a technical description of the implementation of the classification of houses project for the Dutch Cadastre. It serves as a documentation for the software developed during the project.

This software is a prototype of a system that classifies all houses and corresponding adresses in the Netherlands. Every house is assigned one of the categories of Table 1. The rule numbers refer to the classification rules that will be handled later. Notice that the classification types are focussed on houses; all other buildings (factories etc.) are classifed as 'Niet Wonen' (i.e. not a house). The global architecture of the software is given in Figure 1, and consists of two parts: The *first* part, that takes place in the GIS Arc/Info: an overlay is performed between the parcel and the building map layers available at the Dutch Cadastre. Also the ACN coordinates (adresses with an x- and y-coordinated) are linked to the buildings. The results of this overlay, a collection of sub-buildings is copied to the Ingres Database. Here the *second* part of the process, the classification takes place. In the classification, a list classification rules (expressed in SQL) is applied to classify all sub-buildings. The classifications are then propagated to the corresponding ACN coordinates on which some extra classification rules are performed. The result of the complete process, is a classification of sub-buildings and ACN coordindates.

This document is set up as follows: In Section 2 the different data sources that are used in the project are described. In Section 3 the overlay part of the process is described in more detail. Section 4 zooms in on the classification process. In Section 5 details on the different scripts used in the process are given.

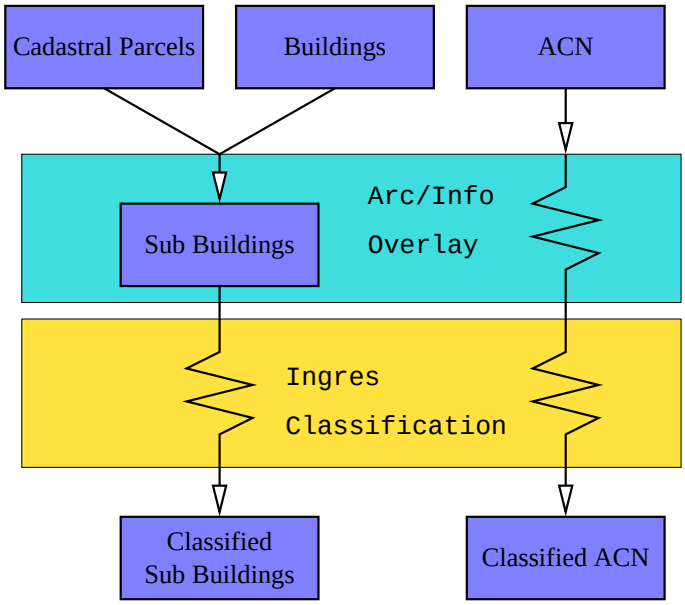

Figure 1: Overall structure of Process

Category	Rule Numbers
Vrijstaand	20, 21, 22, 23, 24, 25, 26
Appartement	10
Rijtjeshuis	50, 51, 52, 53, 54, 55, 56, 57, 58, 59
Eindwoning	40, 41, 42, 43, 44, 45, 46
Twee-onder-een-kap	30, 31, 32, 33, 34, 35, 36, 38, 39
Niet Wonen	-3
Anders	98
Onbekend	-1, -2

Table 1: Classifications and Corresponding Rule Numbers

2 Data Sources

For the classification of houses data from different sources is used. In the following sections a description of the data is given.

2.1 ACN Coordinates

The ACN Coordinates file contains the spatial location of all mail addresses recognized by the dutch PTT. The format of the file is a gzip compressed comma (’,’) separated file with the following attributes:

Attribute	Type
Gemeente	string
Woonplaats NEN	string
Woonplaats PTT	string
Straatnaam NEN	string
Straatnaam PTT	string
Straatnaam	string
Postcode	string
Huisnr	string
Toevoeging	string
Herkomst	string
X-coordinate	integer
Y-coordinate	integer

2.2 Ingres (Query-tool) Database Tables

This classification software is based on the ideas developed for the query-tool system, and cannot function without the underlying database of the querytool up and running. In fact, the classification process is run inside the query-tool database and creates some extra tables there. The base tables and views from the query-tool database that are used are: `mo_recht`, `mo_subject`, `object_parcel` , `akr_object_a` and `akr_objectadres`.

2.3 LKI Dumps

The query-tool database imports its LKI data from Ingres dump files. The analysis software needs to load this data into Arc/Info. Instead of dumping the data from the query-tool, we re-use the dump that was used to load the data into the query-tool. These dump files with names `xfio_line.dat.gz`, `xfio_parcel.dat.gz`, `xfio_boundary.dat.gz` and `xfio_text.dat.gz` are used. Because LKI contains temporal data, a selection is made so that only current data is used. This is the data where (TMAX = 0). Also a selection is made so that only cadastral data is used. This is done by only selecting objects that have SELCO = 'B'.

3 The Overlay Process

This section describes the process that performs the overlay between the different map layers. This process is performed in the Arc/Info GIS software. This process is modeled as in Figure 2. As can be seen, it consists of a big number of scripts that read input files (the top boxes). These files are loaded into Arc/Info layers. On these layers many operations that pre-process the data and perform the overlay are performed (middle of figure). The resulting layers are again dumped to files (bottom part), so that they can be used in the next step of the classification process.

The main operations that are performed in Arc/Info are the following:

[this needs some work]

- The cleanup of the building layer. The buildings in the LKI layer (from the `xfio_line` table are stored as polylines. The overlay operation works on polygons. Cleanup performs the necessary operations for this conversion.
- Construction of the parcel layer from `xfio_boundary` and `xfio_parcel`.
- The actual overlay.
- Point in Polygon tests for the ACN coordinates.

4 The Classification Process

This section describes the classification process. It is performed in the DBMS Ingres. The structure of this process is given in Figure 3.

During the classification the Ingres database is used for two purposes. First, some tables are used retrieve data that is needed for the classification. Second, the classification process is performed inside the database.

Inside the database the following datamodel is used for the classification. An E/R Diagram of the relations is given in Figure 4.

```
create table gebouwen(
   id                integer4,   // Unique building identifier.
```

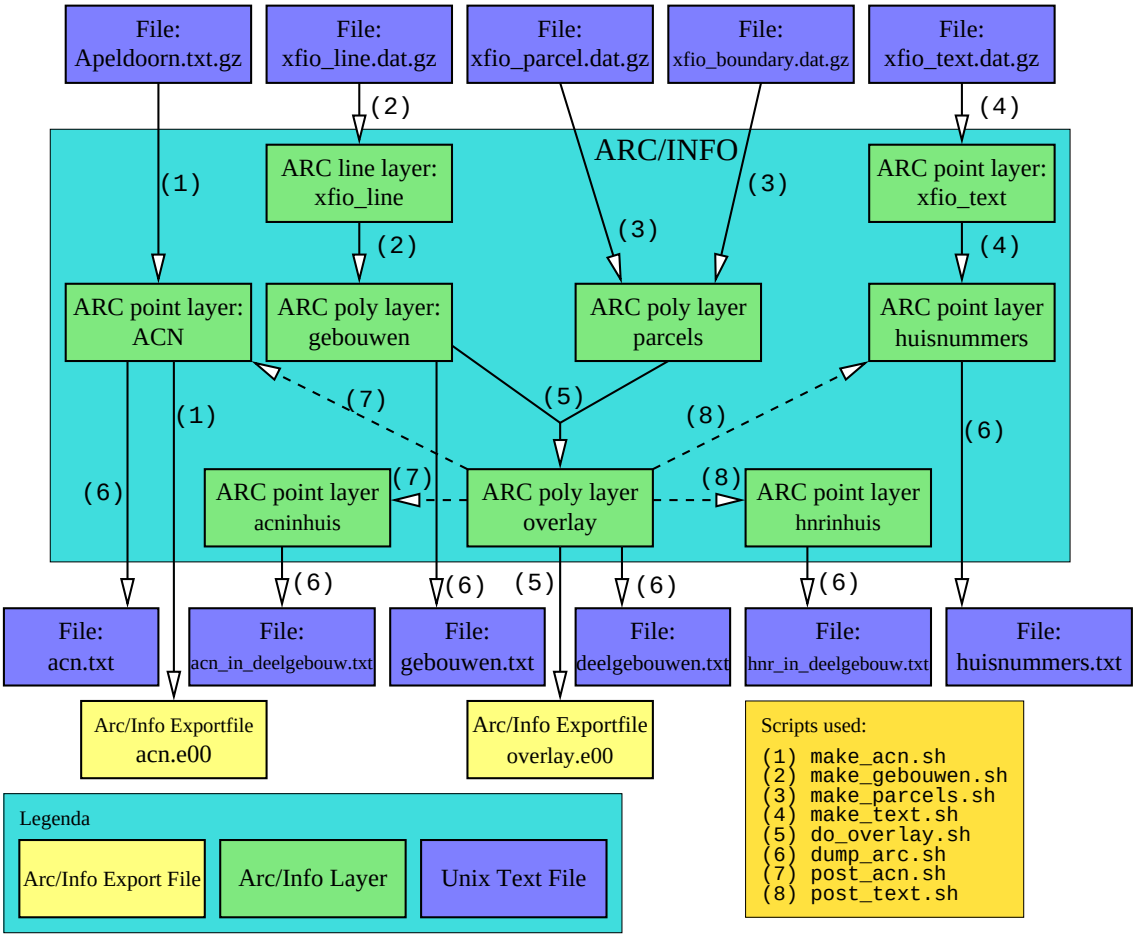

Figure 2: Overlay performed in Arc/Info.

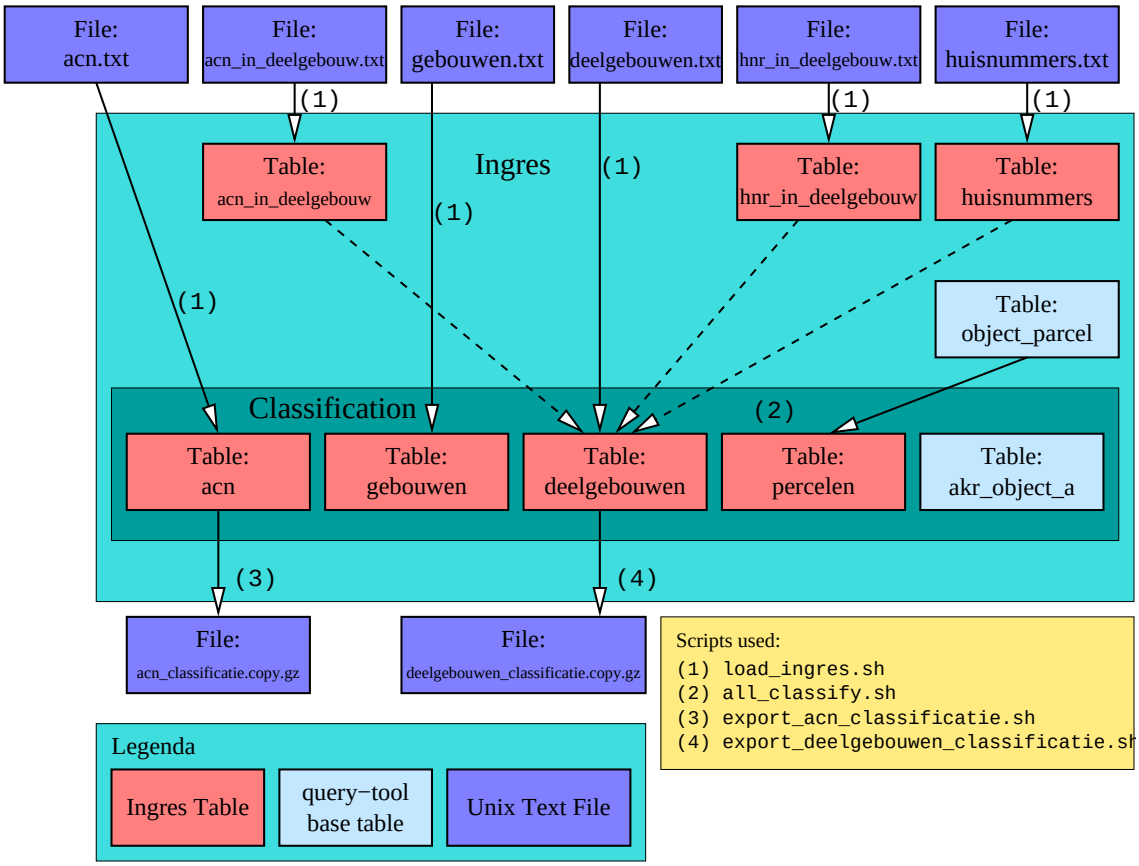

Figure 3: Classification performed in Ingres.

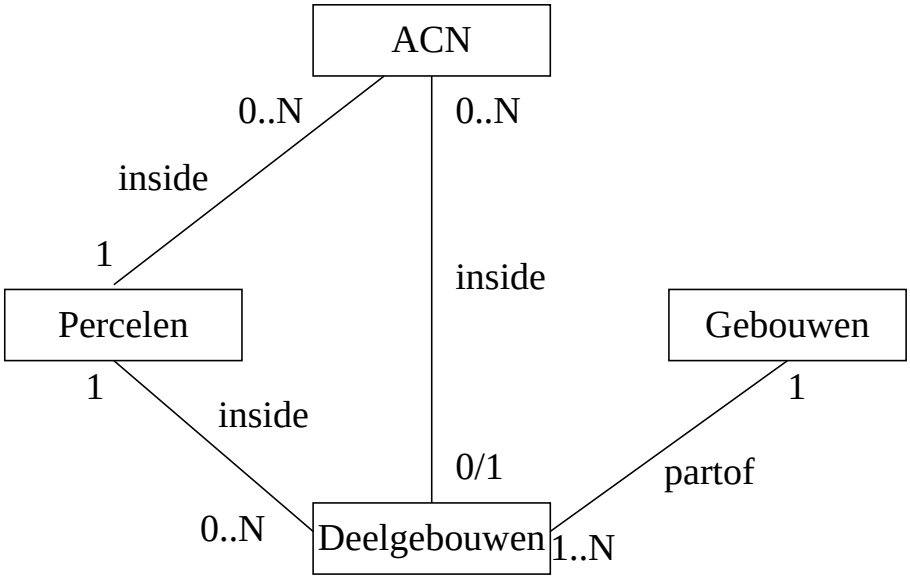

Figure 4: Database model used for classification

```
   area               float8,     // Area of the building in mm^2.
   n_deelgebouwen     integer4,   // Number of sub-buildings.
   n_acn              integer4,   // Number of ACN inside this building.
   n_percelen         integer4,   // Number of parcels below building.
   n_huisnummers      integer4    // Number of house numbers inside this building.
)

create table deelgebouwen(
   id                 integer4,   // Unique partial building identifier.
   area               float8,     // Area of this deelgebouw in mm^2.
   g_akr_objectnummer char(17),   // Parcel id on which this deelgebouw stands.
   gebouwen_id        integer4,   // ID of gebouw of which this deelgebouw is a part.
   n_buren            integer4,   // Number of adjacent deelgebouwen.
   n_acn              integer4,   // Number of acn coordinates inside this deelgebouw.
   n_huisnummers      integer4,   // Number of house numbers insdie this deelgebouw.
   cc_11tot15         char(1),    // 'T' if cultuurcode indicates living ('F' otherwise)
   classificatie      integer4    // House type (assigned during classification).
)

create table acn(
   id                 integer4,   // Unique ID for ACN coordinate.
   plaats             char(40),   // \
   straatnaam         char(40),   // |
   postcode           char(40),   // | Address of this
   huisnummer         integer4,   // | ACN Coordinate
   huistoev           char(5),    // |
   adrtxt             char(10),   // /
   aantal             integer4,   // Number of addresses with on same location.
   g_akr_objectnummer char(17),   // ID of perceel in which ACN is positioned.
   in_deelgebouw      integer4,   // ID of deelgebouw in which ACN is positioned.
                                  // -1 if ACN is not inside a building.
   classificatie      integer4    // House type (assigned during classification).
)
```

4.1 The Classification Rules

The classification of the houses takes place in two phases. In the first phase a classification is performed on the `deelgebouwen` layer. In this phase all deelgebouwen get a classification. These classifications are propagated to the ACN coordinates that belong to the buildings. After the propagation, the second phase starts. In this phase classification rules are performed on the ACN coordinates. Especially ACN coordinates that do not belong to a building are classified in this phase. In the following paragraphs the two classification phases will be discussed.

Classification of Deelgebouwen All classification rules in the project have the same pattern. Below one classification rule is given.

```
update deelgebouwen
from gebouwen,acn,percelen
set classificatie = 43
where
       gebouwen.id = deelgebouwen.gebouwen_id
   and deelgebouwen.g_akr_objectnummer = percelen.g_akr_objectnummer
   and acn.g_akr_objectnummer = percelen.g_akr_objectnummer
   and deelgebouwen.classificatie = -1
   and deelgebouwen.n_buren = 1
   and deelgebouwen.n_acn = 1
   and deelgebouwen.n_huisnummers = 1
   and deelgebouwen.cc_11tot15 = 'T'
   and gebouwen.n_deelgebouwen > 2
   and gebouwen.n_percelen > 2
   and percelen.n_acn < 2;
```

This SQL assigns classification 43 (end of a row) to all deelgebouwen for which the following holds:

- This deelgebouw is not classified yet (has classification -1).
- This deelgebouw had 1 neighbouring deelgebouw.
- This deelgebouw contains 1 acn.
- This deelgebouw contains 1 housenumber.
- This deelgebouw has classification 'wonen'.
- Its associated 'hoofdgebouw' consists of more than 2 deelgebouwen.
- Its associated 'hoofdgebouw' stands on more than two parcels.

Propagation of classifications to ACN After the sub-buildings have been classified, the classifications are propagated to the corresponding ACN coordinates. For every ACN that is inside a sub-building the classification is simply copied. If the ACN is not inside a building it gets classification -2. We use one extra propagation rule: If a coordinate is not in a building, but it is on a parcel with exactly 1 building, we assume the ACN belongs to that building.

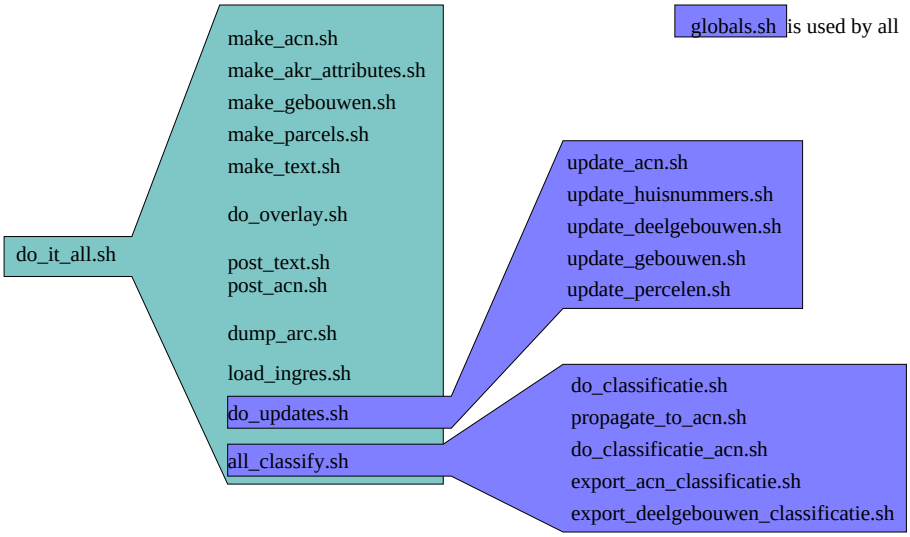

Figure 5: Overview of scripts involved in analysis

Classification of ACN coordinates In the final stage we perform some classifications rules on ACN coordinates that are not related to deelgebouwen. The most important rules are:

- ACN that are on a parcel (without an associated building) where there are appartment-related rights on the parcel, are classified as appartment.
- Unclassified ACN that are in a building where some addresses are classified as 'Rijtjeshuis' also get the classification 'Rijtjeshuis' (since they usually come in groups).
- In buildings of 'Rijtjeshuizen' the ACN with the highest and lowest house number get the classification 'Eindwoning'. This case happens when we have a rental building with 'Rijtjeshuizen'. Because there is no geometry of the separate buildings, we assume that the adresses with the lowest and highest house numbers are at the end.

5 Scripts

The whole process of conversion, overlay and classification is performed by a series of scripts. This section contains an overview of all scripts that are used in the process, together with a short description of the script. Figure 5 gives a call-graph of all the scripts.

5.1 do_it_all.sh

This script is the master script. If all pre-conditions are OK and all parameters in 'globals.sh' are set correctly, the scripts creates some files with the resulting classifications.

Uses All other scripts.

5.2 globals.sh

This file contains all global parameters that influence the conversion process. Below you can find an overview of the parameters.

$RESULTDIR The result of the classification process is a set of files. The files are stored in this directory.

$DB This is the name of the Ingres database which is used in the classification. It is assumed that the database contains all tables belonging to the IQT system.

$TMPDIR In this directory temporary files are stored.

$ACNFILE This variable contains the filename in which the ACN coordinates are stored. This file is compressed with the gzip utility.

$LKIDIR This directory should contain gzip compressed export files of the LKI system. The files in this directory should have names like 'xfio_text.dat.gz'

$ARCDIR This is the working directory for Arc/Info. In this directory all Arc/Info layers are created.

$MAXAREA During the cleanup of the overlay, all polygons with an area smaller than a given area $MAXAREA are deleted. This area is given in mm^2. A good value for this variable is 20000000 $(20m^2)$. But because Arc/Info sometimes crashes on the value, it is parameterized.

Is used by: Virtually all scripts.

5.3 make_acn.sh

This script loads the ACN data into an Arc/Info coverage. The original ACN data is supposed to resized in a file. It is supposed that this file is compressed with gzip, and the name of the file can be found in $ACNFILE. The name of the created coverage is 'acn'.

Uses: acn_to_arc.perl, acn_to_info.perl

5.4 make_gebouwen.sh

This script converts the line information from the LKI dumps into an arc layer called 'gebouwen'.

Uses: process_xfio_line.perl, xfio_line_to_info.perl

5.5 make_parcels.sh

This script converts the xfio_parcel and xfio_boundary tables from the LKI dumps into an Arc/Info layer called 'parcels'.

Uses: process_xfio_parcel.perl, process_xfio_boundary.perl, xfio_parcel_to_info.perl

5.6 make_text.sh

This script converts the xfio_text table from the LKI dumps into an Arc/Info layer called 'xfio_text'.

Uses: process_xfio_text.perl

5.7 do_overlay.sh

The script computes the overlay between the Arc layers 'parcels' and 'gebouwen'. The result is a new Arc layer called 'overlay'. This script performs the following steps.

- With the 'intersect' command the parcels and gebouwen layer are overlayed.
- From this overlay all polygons with an area less than (currently) 20 square meters are eliminated. This means that these polygons are merged with the biggest neighbouring polygon.
- With the 'unsplit' command all pseudo nodes (i.e. nodes with two outgoing arc are removed.
- With the 'build' command proper line and polygon coverages are maded from the overlay.
- All dangling lines (lines that have the same polygon to the left and to the right) are purged.
- With a separate script ('house_frequency.perl') the number of neigbours for every house is calculated. This info is added as the attribute 'neighbourcount' to the overlay layer. The house_frequency.perl script is described next.
- Finally the 'overlay' coverage is exported for use in different packages.

Uses: house_frequency.perl

5.8 house_frequency.perl

The number of neighbours of a house is defined by the number of internal walls that a house has. This is calculated in the following way:

- In the script 'do_overlay.sh' a table is made with all the internal walls of buildings. These are walls that have a different house to the left and to the right.
- These internal walls are dumped to a file called 'lines.copy'.
- This file is input to a perl script 'house_frequency.perl' that calculates the number of internal walls for every house.
- This count is dumped to a file 'frequency.copy', and reloaded to Arc/Info where it is joined again to the 'overlay' table.

5.9 post_text.sh

This script select all house number labels from the 'xfio_text' layer, and finds out in which parcel and building the number is positioned. The result of the script is a new Arc/Info layer called 'huisnummers'.

- All text items with lki code 202 are selected. This are the items which are house numbers.
- The Arc/Info intersect command is used to find the parcel and the building in which every house number is positioned. Ideally every house number is positioned inside building, but sometimes the number is placed outside a building, in this case the parcel can be used to find out to which building the house number belongs.

5.10 post_acn.sh

This script finds the corresponding parcel and building for every ACN coordinate.

- With the Arc/Info an intersect operation a parcel is found for every ACN coordinate. The 'g_akr_objectnummer' is added as a new attribute to the acn layer.
- Another intersect operation finds all ACN coordinates that are positioned inside buildings. These coordinates are put in an new layer 'acninhuis'.

5.11 dump_arc.sh

This scripts dumps all relevant tables from Arc/Info to files. These files can later be imported by Ingres. The following layers and attributes are dumped.

Coverage name	Export file
overlay	deelgebouwen.txt
acn	acn.txt
huisnummers	huisnummers.txt
gebouwen	gebouwen.txt
acninhuis	acn_in_deelgebouw.txt
hnrinhuis	hnr_in_deelgebouw.txt

5.12 load_ingres.sh

Load the the dumps made by 'dump_arc.sh' into the ingres database. The following steps are taken:

- Tables are created in Ingres.
- With the copyrel command the tables are filled.
- Indices are built on the tables.

Table	Attribute	Value
acn	in_deelgebouw	Reference to the sub-building inside which this acn is located. (-1 if not inside building)
deelgebouwen	n_huisnummers	Set attribute to the number of house numbers inside this sub-building
deelgebouwen	n_acn	Set attribute to the number of ACN coordinates inside this sub-building
deelgebouwen	cc_11tot15	This code is 'T' if the culture code of the objectaddress belonging to this building is between 11 and 15
gebouwen	n_percelen	Set to the number of parcels on which this building stands
gebouwen	n_huisnummers	Set to the number of house numbers inside this building
gebouwen	n_deelgebouwen	Count the number of sub-buildings of this building
gebouwen	n_acn	This is the number of ACN coordinates inside the building
huisnummers	in_deelgebouw	Set value to sub-building in which huisnummer is located
percelen	g_akr_objectnummer	Parcel number
percelen	n_gebouwen	Number of different buildings on this parcel
percelen	n_acn	Number of ACN Coordinates on this parcel
percelen	n_huisnummers	Number of house number strings on this parcel
percelen	cultuurcode	Culture code for parcels with code in: 0,11,12,13,14,15 All other parcels get '- -'

Figure 6: Attributes that are updated by the Update Scripts

5.13 do_updates.sh

For the classification various aggregate values are needed in the tables. These values are calculated with the update scripts. The scripts also initialize different attributes, for example all buildings get the initial status unclassified. All in all the followings attributes are set:

Uses: update_huisnummers.sh, update_deelgebouwen.sh, update_gebouwen.sh, update_percelen.sh

5.14 all_classify.sh

This is the master classification script. It calls all other scripts in the correct order.

Uses: do_classificatie.sh, propagate_to_acn.sh, do_classificatie_acn.sh

5.15 do_classificatie.sh

This script performs the classification on sub-buildings. When this script is finished, the attribute 'classificatie' of the table 'deelgebouwen' is filled. Non classified buildings get code '-1'.

5.16 propagate_to_acn.sh

This script propagates the classification of the sub-buildings to the ACN coordinates table. First ACN coordinates which are inside a building get their classification copied. Then ACN coordinates that are on a parcel with exactly 1 building get the classification of the building. ACN coordinates that do not have an associated building get classification -2.

5.17 do_classificatie_acn.sh

This script tries to find a classification for ACN that do not have a corresponding building. Among others, it contains the following rules:

- ACN coordinates on a parcel with appartment indication are classified as appartments.
- Unclassified ACN coordinates inside a building which has been classified as a 'rijtjeshuis' get the classification rijtjeshuis.
- A 'rijtjeshuis' where the housenumber equals the maximum or minimum house number for that building get classification 'eindwoning'.

6 Conclusions

This document describes the prototype implementation for the classification of houses on the Cadastral Map. As is proven by this piece of software is is possible to classify many of the houses on the map. However, the current software is very complex and depends on the availability of too many packages (Ingres,Arcview,Query-tool). The reason for the complex nature of this prototype is that it was developed with as much re-use of existing software and know-how as possible. A less complex and more efficient implementation is definitely feasable.