AUTOMATION WALHALLA: A SHOWCASE OF AUTOMATING COMPLEX PROCESSES

SESSION 2020

Tuesday 17 October 2017

PRESENTER

Hylke Hoekstra

Functional Application Manager

Leiden University

h.r.hoekstra@sea.leidenuniv.nl

Universiteit Leiden
The Netherlands

In Leiden and The Hague

Universiteit Leiden
The Netherlands


7
Faculties
15
Nobel Prizes
46
Bachelor's programmes
78
Master's programmes
115
Nationalities
26,900
Students
104,000
Alumni
2
Cities
5,500
Employees

IN FIGURES

VERSION

PS CS 9.0 (9.2 upgrade planned 2018)

Peopletools 8.55.13 bundle 41

Live since 2010 and continuously evolving

10 FTE functional team

Collaborating Higher Ed institutions: Leiden University, University of Amsterdam, Hogeschool van Amsterdam (UoAS)

Technical partner and hosting:

OVERVIEW OF THIS PRESENTATION

1. DEREGISTRATION IN THE NETHERLANDS
   A brief overview of the rules and procedures
2. VISUALISING THE PROCESS FLOW
   How we visualized the process flow (multiple times)
3. CALCULATIONS WITH THE HELP OF SQL VIEWS
   How SQL views can help in (complex) calculations

DEREGISTRATION IN THE NETHERLANDS

A brief overview of the key rules and procedures

(DE)REGISTRATION VIA STUDIELINK

- Studielink is a national web portal for student registration at all higher education institutions in The Netherlands

- Every student gets his or her own account which can be used for registration and deregistration

- Institutions control when a student can request registration

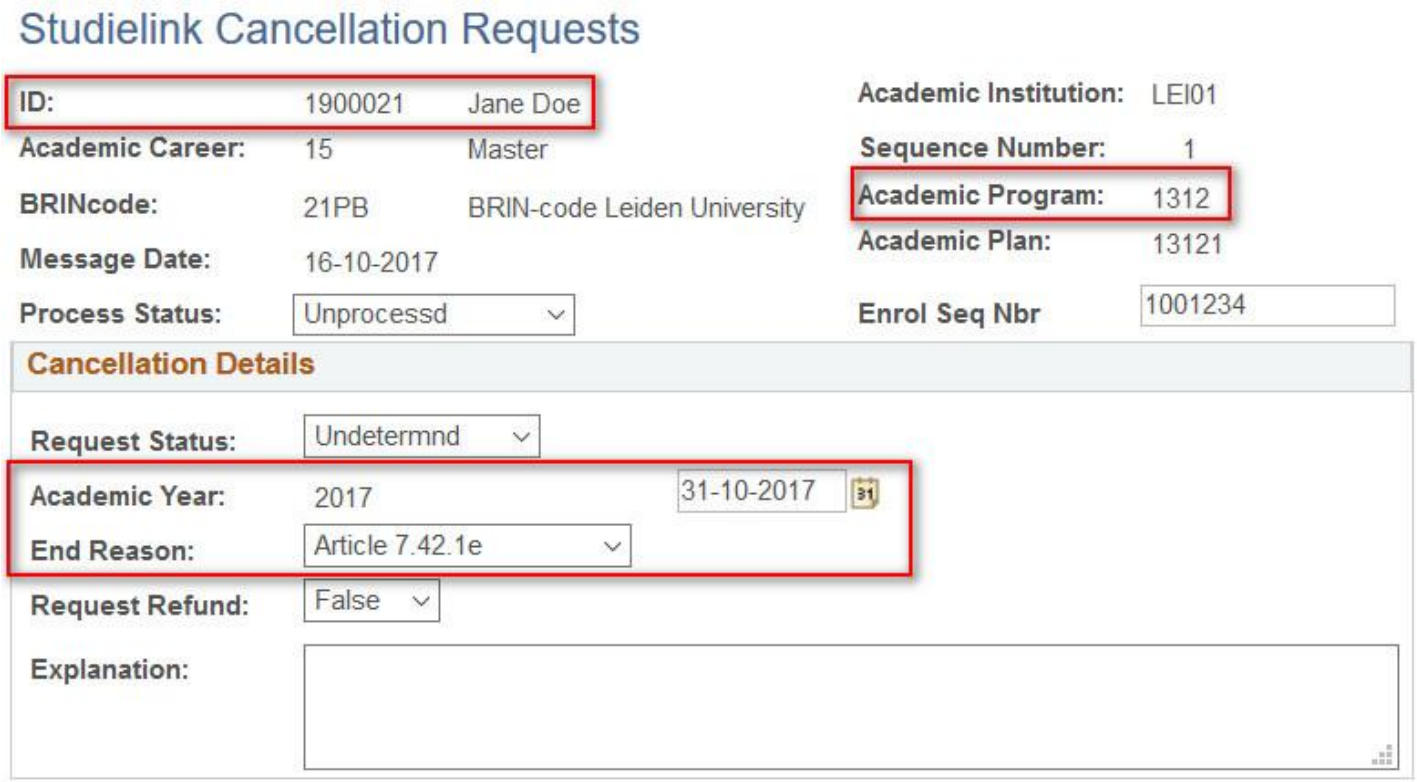
- Studielink controls when a student can request deregistration

CRITERIA FOR REQUESTING DEREGISTRATION

- Studielink uses three criteria to allow students to request deregistration:
  - The student needs to have a definitive registration
  - Deregistration is only possible per the last day of the month (31st, 30th, 28th, 29th)
  - A student can only request deregistration per the end of the current month or later (past dates are not allowed)

STORING THE DEREGISTRATION REQUESTS

Requests for deregistration submitted via Studielink are stored in a special *deregistration table* in PeopleSoft Campus Solutions (uSis).



TUITION FEES (SIMPLIFIED)

Two types of tuition fees:

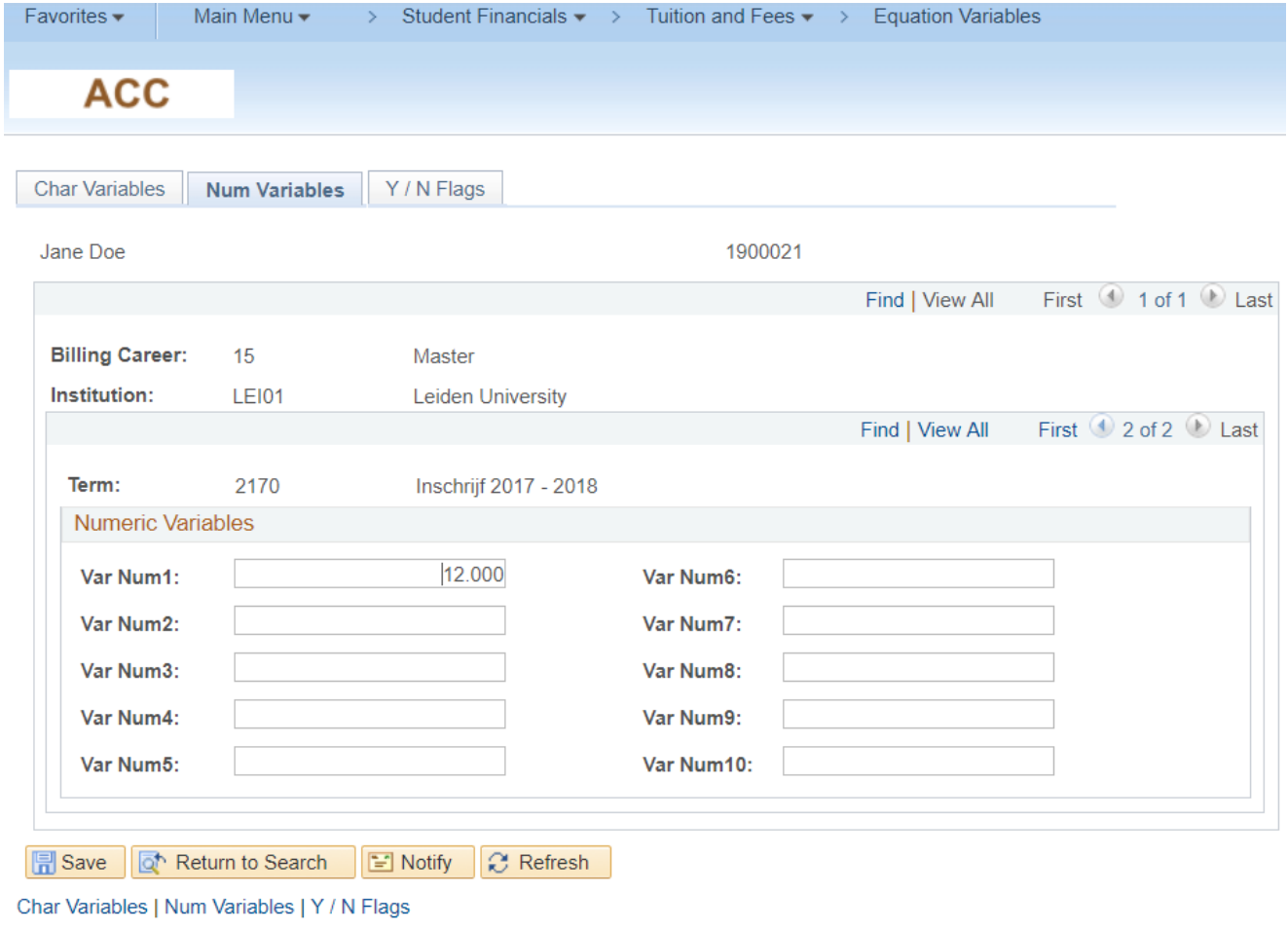
1. Statutory fee

2. Institutional fee

Students pay per month that they are registered (irrespective the number of registrations)

We will not discuss all exceptions

USING EQUATION VARIABLES FOR TUITION FEE CALCULATION



MANUALLY DEREGISTERING A STUDENT

On average the manual deregistration process takes up to 5 minutes per student deregistration

There were approximately 4000 manual deregistrations per year

In January and February there are over 200 deregistration requests per week

It would take one person at least 16-20 hours per week to manually process all these requests in those months

Students can wait up to six weeks before their request is processed

Looks like a genuine business case; can we automate this process?

VISUALISING THE PROCESS FLOW

A description of the iterations of visualizing the process

INQUIRY AT STUDENT ADMINISTRATION

We asked the Student Administration Department to tell us about the deregistration process

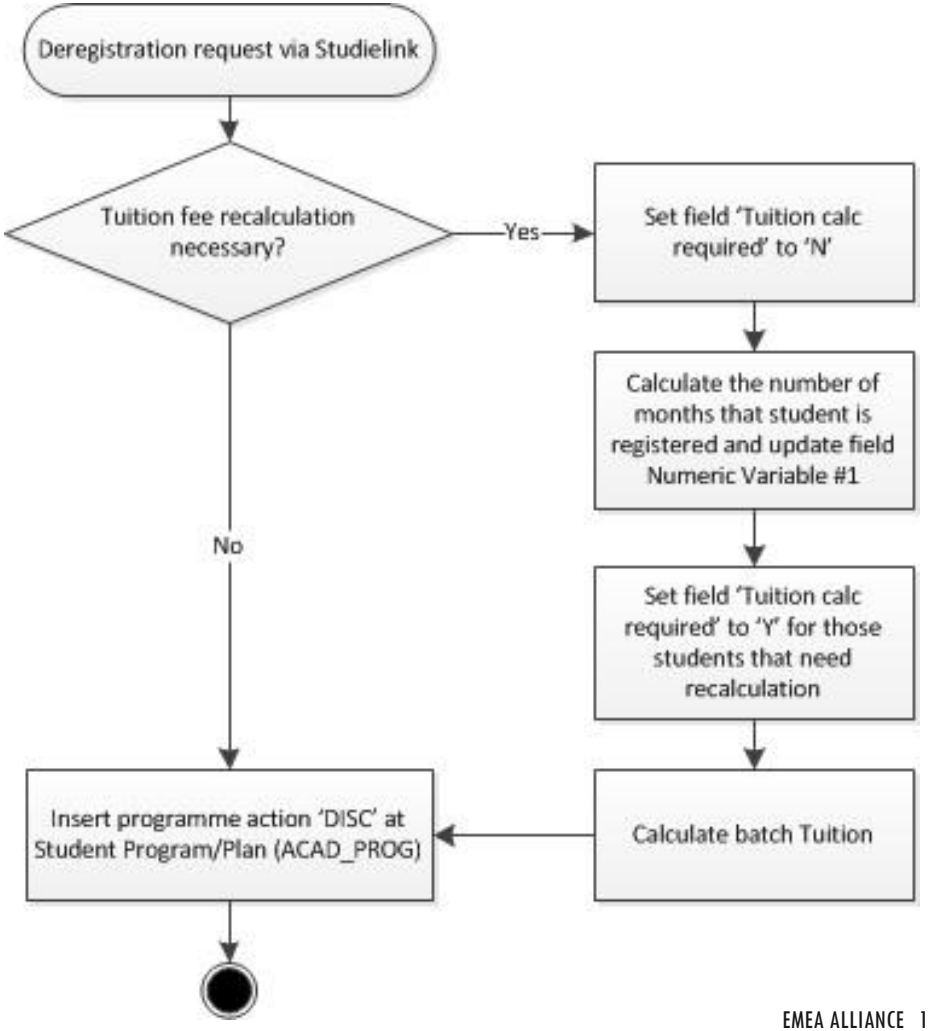
- They found it difficult to make a list of requirements

- Basic procedure:

➢ Check if recalculation is necessary

➢ Add correct programme action.

BASIC PROCESS FLOW (2016)



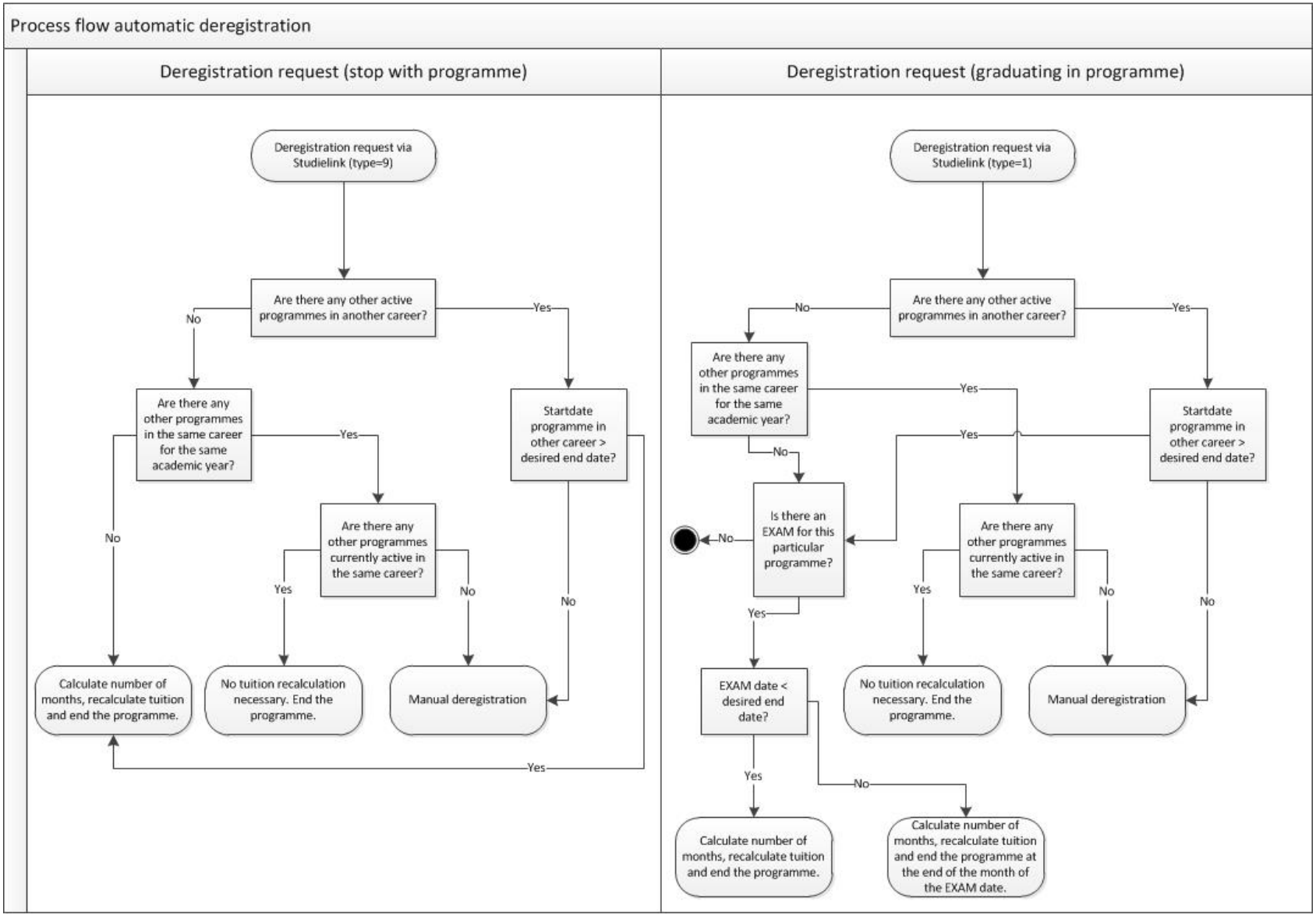
FUNCTIONAL REQUIREMENTS

Calculation is necessary when

- There are no other programmes currently active within the same career
- There are no other programmes currently active in another career with a starting date that is earlier than the desired termination date

In case of graduation:

- Only terminate registration if there is an EXAM registered
- Always terminate the registration 'after' the EXAM date


Process flow automatic deregistration
Deregistration request (stop with programme)
Deregistration request via Studielink (type=9)
Are there any other active programmes in another career?
No
Yes
Are there any other programmes in the same career for the same academic year?
Yes
No
Are there any other programmes currently active in the same career?
Yes
No
Startdate programme in other career > desired end date?
No
Calculate number of months, recalculate tuition and end the programme.
No tuition recalculation necessary. End the programme.
Manual deregistration
Yes
Deregistration request (graduating in programme)
Deregistration request via Studielink (type=1)
Are there any other active programmes in another career?
No
Yes
Are there any other programmes in the same career for the same academic year?
Yes
No
Is there an EXAM for this particular programme?
No
Yes
Yes
Are there any other programmes currently active in the same career?
Yes
No
Startdate programme in other career > desired end date?
No
EXAM date < desired end date?
Yes
No
No tuition recalculation necessary. End the programme.
Manual deregistration
Yes
No
Calculate number of months, recalculate tuition and end the programme.
Calculate number of months, recalculate tuition and end the programme at the end of the month of the EXAM date.

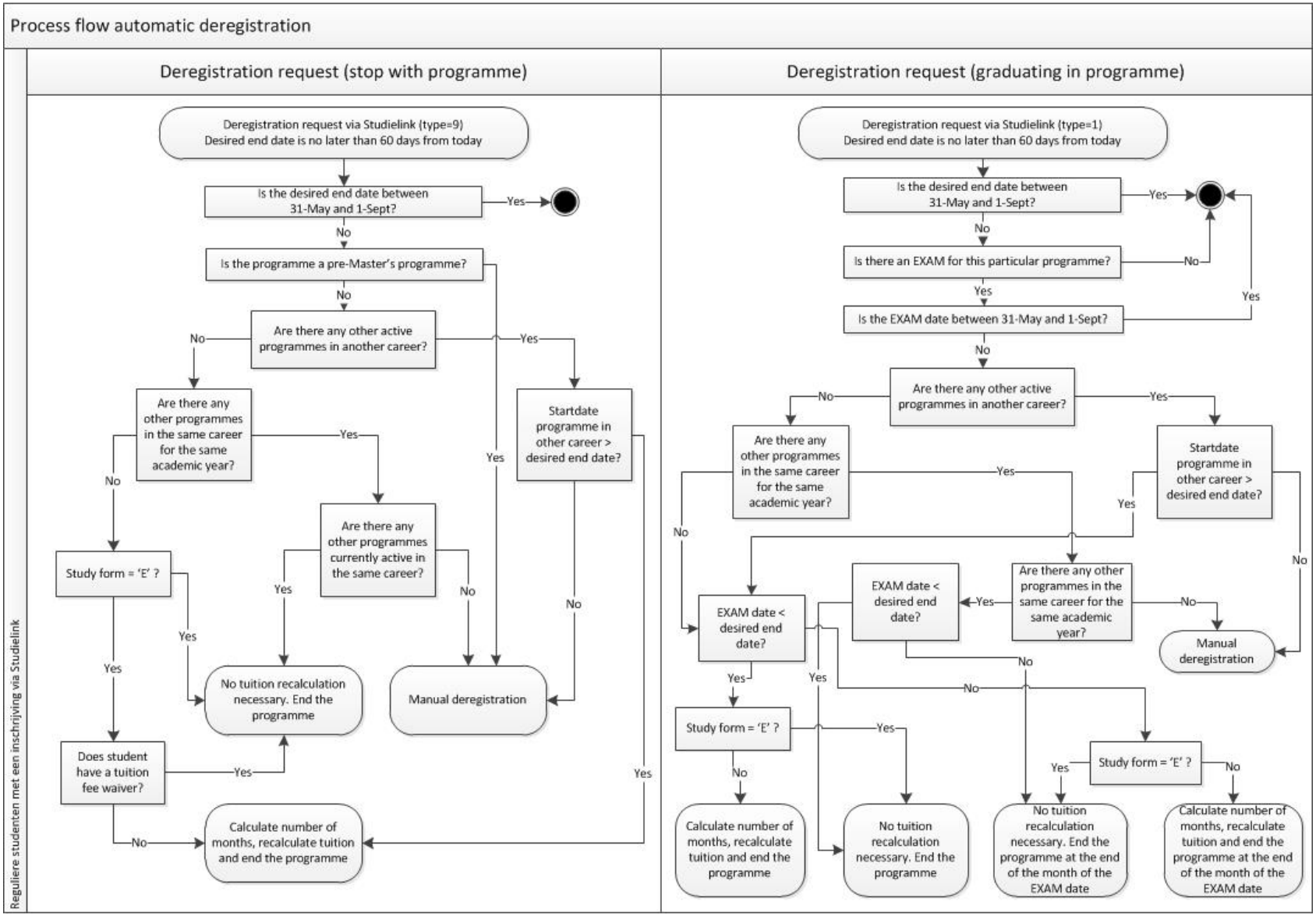
ADDITIONAL FUNCTIONAL REQUIREMENTS

Do not terminate the registration if the desired termination date is greater than 31st of May

Do not terminate the registration if it concerns a Pre-Master's Programme

Do not recalculate the tuition fee if the Study Form = 'E'

Do not recalculate the tuition fee if we have received a waiver


Process flow automatic deregistration
Reguliere studenten met een inschrijving via Studielink
Deregistration request (stop with programme)
Deregistration request via Studielink (type=9)
Desired end date is no later than 60 days from today
Is the desired end date between 31-May and 1-Sept?
Yes
No
Is the programme a pre-Master's programme?
No
Yes
Are there any other active programmes in another career?
No
Yes
Are there any other programmes in the same career for the same academic year?
No
Yes
Are there any other programmes currently active in the same career?
Yes
No
Study form = 'E' ?
Yes
Yes
No
Startdate programme in other career > desired end date?
Yes
No
No tuition recalculation necessary. End the programme
Manual deregistration
Does student have a tuition fee waiver?
Yes
No
Calculate number of months, recalculate tuition and end the programme
Deregistration request (graduating in programme)
Deregistration request via Studielink (type=1)
Desired end date is no later than 60 days from today
Is the desired end date between 31-May and 1-Sept?
Yes
No
Is there an EXAM for this particular programme?
No
Yes
Is the EXAM date between 31-May and 1-Sept?
Yes
No
Are there any other active programmes in another career?
No
Yes
Are there any other programmes in the same career for the same academic year?
No
Yes
Startdate programme in other career > desired end date?
Yes
No
EXAM date < desired end date?
No
Yes
Are there any other programmes in the same career for the same academic year?
Yes
No
Manual deregistration
EXAM date < desired end date?
Yes
No
Yes
Study form = 'E' ?
Yes
No
Study form = 'E' ?
Yes
No
Calculate number of months, recalculate tuition and end the programme
No tuition recalculation necessary. End the programme
No tuition recalculation necessary. End the programme at the end of the month of the EXAM date
Calculate number of months, recalculate tuition and end the programme at the end of the month of the EXAM date

MORE ADDITIONAL FUNCTIONAL REQUIREMENTS

Additional rules concerning a ‘special’ programme (1298)

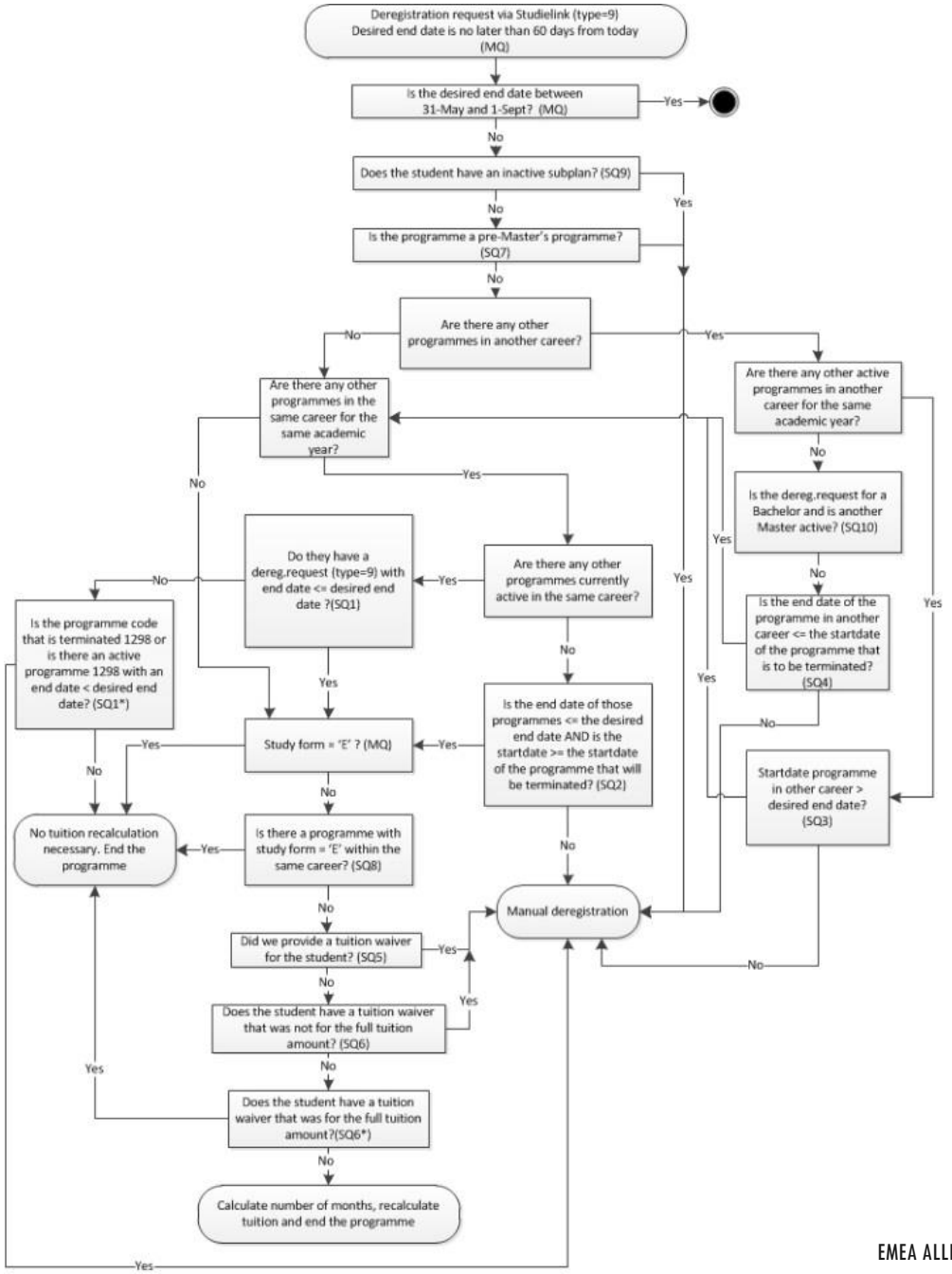
Additional rules concerning tuition fee waivers

TECHNICAL CONSTRAINTS/REQUIREMENTS

App Engine cannot insert row on Student Program/Plan if the student has an inactive subplan

Batch tuition calculation cannot calculate if there is an inactive program within the same academic year in a career with a lower career number

Tuition fee waivers cannot be recalculated automatically


Deregistration request via Studielink (type=9)
Desired end date is no later than 60 days from today
(MQ)
Is the desired end date between 31-May and 1-Sept? (MQ)
Yes
No
Does the student have an inactive subplan? (SQ9)
Yes
No
Is the programme a pre-Master's programme? (SQ7)
Yes
No
Are there any other programmes in another career?
No
Yes
Are there any other programmes in the same career for the same academic year?
No
Yes
Are there any other active programmes in another career for the same academic year?
No
Is the dereg.request for a Bachelor and is another Master active? (SQ10)
Yes
No
Is the end date of the programme in another career <= the startdate of the programme that is to be terminated? (SQ4)
Yes
No
Startdate programme in other career > desired end date? (SQ3)
Yes
No
Are there any other programmes currently active in the same career?
Yes
No
Is the end date of those programmes <= the desired end date AND is the startdate >= the startdate of the programme that will be terminated? (SQ2)
Yes
No
Do they have a dereg.request (type=9) with end date <= desired end date ?(SQ1)
Yes
No
Is the programme code that is terminated 1298 or is there an active programme 1298 with an end date < desired end date? (SQ1*)
Yes
No
Study form = 'E' ? (MQ)
Yes
No
Is there a programme with study form = 'E' within the same career? (SQ8)
Yes
No tuition recalculation necessary. End the programme
No
Manual deregistration
Did we provide a tuition waiver for the student? (SQ5)
Yes
No
Does the student have a tuition waiver that was not for the full tuition amount? (SQ6)
Yes
No
Does the student have a tuition waiver that was for the full tuition amount?(SQ6*)
Yes
No
Calculate number of months, recalculate tuition and end the programme
Yes

RESULT:
A VERY COMPLEX QUERY FOR THIS PROCESS

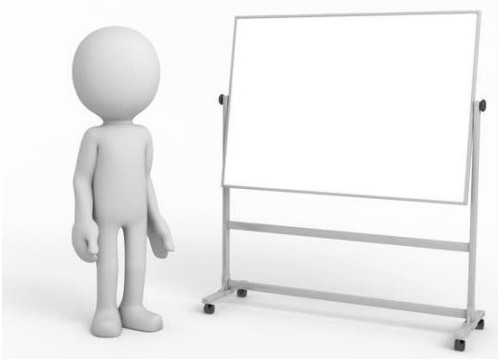
```
SELECT DISTINCT B.EMPLID, B.ACAD_CAREER, B.STDNT_CAR_NBR, ROUND((TO_DATE(
TO_CHAR(A.SSR_SL_END_DT_NLD,'YYYY-MM-DD')) - TO_DATE( TO_CHAR(D.EFFDT,'YYYY-MM-
DD'))) / (365/12)), T.STRM, A.ACAD_PROG
  FROM PS_SSR_SL_WDRW_NLD A, PS_ACAD_PROG B, PS_SSR_STD_PRG_NLD C,
PS_SSR_STD_PRG_NLD D, PS_STDNT_CAR_TERM T, PS_STDNT_EQUTN_VAR X
  WHERE ( A.SSR_SL_END_RSN_NLD = '9'
    AND A.SSR_SL_WPROCST_NLD = 'U'
    AND A.SSR_SL_WRGSTST_NLD = 'U'
    AND A.ACAD_CAREER IN ('10','15','20')
    AND A.SSR_SL_ACAD_YR_NLD = '20' || SUBSTR( T.STRM,2,2)
    AND A.SSR_SL_END_DT_NLD <= TO_DATE(TO_CHAR(SYSDATE,'YYYY-MM-DD'),'YYYY-MM-
DD')+60
    AND A.SSR_SL_END_DT_NLD < TO_DATE('01-06-' || TO_CHAR(
A.SSR_SL_ACAD_YR_NLD+1),'DD-MM-YYYY')
    AND A.EMPLID = B.EMPLID
    AND A.INSTITUTION = B.INSTITUTION
    AND A.ACAD_CAREER = B.ACAD_CAREER
    AND A.ACAD_PROG = B.ACAD_PROG
    AND B.EFFDT = (
      (SELECT MAX(B_ED.EFFDT) FROM PS_ACAD_PROG B_ED
        WHERE B.EMPLID = B_ED.EMPLID
          AND B.ACAD_CAREER = B_ED.ACAD_CAREER
          AND B.STDNT_CAR_NBR = B_ED.STDNT_CAR_NBR)
    AND B.EFFSEQ =
        (SELECT MAX(B_ES.EFFSEQ) FROM PS_ACAD_PROG B_ES
        WHERE B.EMPLID = B_ES.EMPLID
          AND B.ACAD_CAREER = B_ES.ACAD_CAREER
          AND B.STDNT_CAR_NBR = B_ES.STDNT_CAR_NBR
          AND B.EFFDT = B_ES.EFFDT)
    AND B.PROG_STATUS = 'AC'
    AND B.EMPLID = C.EMPLID
    AND B.ACAD_CAREER = C.ACAD_CAREER
    AND B.STDNT_CAR_NBR = C.STDNT_CAR_NBR
    AND C.SSR_PRG_ACTION_NLD = 'DEFT'
    AND C.ACAD_YEAR = A.SSR_SL_ACAD_YR_NLD
    AND C.SAD_SL_ENRLSQ_NLD = A.SAD_SL_ENRLSQ_NLD
    AND C.SSR_FORM_STUDY_NLD = 'S'
    AND B.EMPLID = D.EMPLID
    AND B.ACAD_CAREER = D.ACAD_CAREER
    AND B.STDNT_CAR_NBR = D.STDNT_CAR_NBR
    AND D.SSR_PRG_ACTION_NLD IN ('MATR','RENR')
    AND D.ACAD_YEAR = A.SSR_SL_ACAD_YR_NLD
    AND NOT EXISTS (SELECT 'X'
FROM PS_ACAD_PROG E
  WHERE E.EFFDT =
      (SELECT MAX(E_ED.EFFDT) FROM PS_ACAD_PROG E_ED
        WHERE E.EMPLID = E_ED.EMPLID
          AND E.ACAD_CAREER = E_ED.ACAD_CAREER
          AND E.STDNT_CAR_NBR = E_ED.STDNT_CAR_NBR)
    AND E.EFFSEQ =
      (SELECT MAX(E_ES.EFFSEQ) FROM PS_ACAD_PROG E_ES
        WHERE E.EMPLID = E_ES.EMPLID
          AND E.ACAD_CAREER = E_ES.ACAD_CAREER
          AND E.STDNT_CAR_NBR = E_ES.STDNT_CAR_NBR
          AND E.EFFDT = E_ES.EFFDT)
    AND E.EMPLID = B.EMPLID
    AND E.ACAD_CAREER = B.ACAD_CAREER
    AND E.STDNT_CAR_NBR <> B.STDNT_CAR_NBR
    AND E.PROG_STATUS = 'AC'
    AND NOT EXISTS (SELECT 'X'
FROM PS_SSR_STD_PRG_NLD F, PS_SSR_SL_WDRW_NLD G
  WHERE F.EMPLID = E.EMPLID
    AND F.ACAD_CAREER = E.ACAD_CAREER
    AND F.STDNT_CAR_NBR = E.STDNT_CAR_NBR
    AND F.SSR_PRG_ACTION_NLD IN ('MATR','RENR')
    AND F.ACAD_YEAR = A.SSR_SL_ACAD_YR_NLD
    AND F.EMPLID = G.EMPLID
    AND F.ACAD_CAREER = G.ACAD_CAREER
    AND G.SCC_BRINCD_NLD = F.SCC_BRINCD_NLD
    AND G.SAD_SL_ENRLSQ_NLD = F.SAD_SL_ENRLSQ_NLD
    AND G.SSR_SL_ACAD_YR_NLD = A.SSR_SL_ACAD_YR_NLD
    AND G.SSR_SL_WPROCST_NLD = 'U'
    AND G.SSR_SL_WRGSTST_NLD = 'U'
    AND G.SSR_SL_END_RSN_NLD = '9'
    AND (( G.SSR_SL_END_DT_NLD < A.SSR_SL_END_DT_NLD
    AND G.ACAD_PROG <> '1298')
    OR G.SSR_SL_END_DT_NLD = A.SSR_SL_END_DT_NLD)))
    AND NOT EXISTS (SELECT 'X'
FROM PS_ACAD_PROG H, PS_SSR_STD_PRG_NLD I
WHERE H.EFFDT =
      (SELECT MAX(H_ED.EFFDT) FROM PS_ACAD_PROG H_ED
        WHERE H.EMPLID = H_ED.EMPLID
          AND H.ACAD_CAREER = H_ED.ACAD_CAREER
          AND H.STDNT_CAR_NBR = H_ED.STDNT_CAR_NBR)
    AND H.EFFSEQ =
      (SELECT MAX(H_ES.EFFSEQ) FROM PS_ACAD_PROG H_ES
        WHERE H.EMPLID = H_ES.EMPLID
          AND H.ACAD_CAREER = H_ES.ACAD_CAREER
          AND H.STDNT_CAR_NBR = H_ES.STDNT_CAR_NBR
          AND H.EFFDT = H_ES.EFFDT)
    AND H.EMPLID = B.EMPLID
    AND H.ACAD_CAREER = B.ACAD_CAREER
    AND H.STDNT_CAR_NBR <> B.STDNT_CAR_NBR
    AND H.PROG_ACTION IN ('DISC','VDIS','BPND')
    AND H.EMPLID = I.EMPLID
    AND H.ACAD_CAREER = I.ACAD_CAREER
    AND H.STDNT_CAR_NBR = I.STDNT_CAR_NBR
    AND I.SSR_PRG_ACTION_NLD IN ('MATR','RENR')
    AND I.ACAD_YEAR = A.SSR_SL_ACAD_YR_NLD
    AND ( I.EFFDT < D.EFFDT
    OR H.EFFDT > A.SSR_SL_END_DT_NLD)
    AND H.ACAD_PROG <> '1298')
    AND NOT EXISTS (SELECT 'X'
FROM PS_ACAD_PROG J, PS_SSR_STD_PRG_NLD K
WHERE J.EFFDT =
      (SELECT MAX(J_ED.EFFDT) FROM PS_ACAD_PROG J_ED
        WHERE J.EMPLID = J_ED.EMPLID
          AND J.ACAD_CAREER = J_ED.ACAD_CAREER
          AND J.STDNT_CAR_NBR = J_ED.STDNT_CAR_NBR)
    AND J.EFFSEQ =
      (SELECT MAX(J_ES.EFFSEQ) FROM PS_ACAD_PROG J_ES
        WHERE J.EMPLID = J_ES.EMPLID
          AND J.ACAD_CAREER = J_ES.ACAD_CAREER
          AND J.STDNT_CAR_NBR = J_ES.STDNT_CAR_NBR
          AND J.EFFDT = J_ES.EFFDT)
    AND J.EMPLID = B.EMPLID
    AND J.ACAD_CAREER <> B.ACAD_CAREER
    AND J.ACAD_CAREER IN ('10','15','20','25')
    AND J.PROG_STATUS = 'AC'
    AND J.EMPLID = K.EMPLID
    AND J.ACAD_CAREER = K.ACAD_CAREER
    AND J.STDNT_CAR_NBR = K.STDNT_CAR_NBR
    AND K.ACAD_YEAR = A.SSR_SL_ACAD_YR_NLD
    AND K.SSR_PRG_ACTION_NLD IN ('MATR','RENR')
    AND K.EFFDT < A.SSR_SL_END_DT_NLD)
    AND NOT EXISTS (SELECT 'X'
FROM PS_ACAD_PROG L, PS_SSR_STD_PRG_NLD M
WHERE L.EMPLID = B.EMPLID
    AND L.ACAD_CAREER <> B.ACAD_CAREER
    AND L.ACAD_CAREER IN ('10','15','20','25')
    AND L.PROG_ACTION IN ('DISC','VDIS')
    AND L.EMPLID = M.EMPLID
    AND L.ACAD_CAREER = M.ACAD_CAREER
    AND L.STDNT_CAR_NBR = M.STDNT_CAR_NBR
    AND M.SSR_PRG_ACTION_NLD IN ('MATR','RENR')
    AND M.ACAD_YEAR = A.SSR_SL_ACAD_YR_NLD
    AND L.EFFDT > D.EFFDT)
    AND NOT EXISTS (SELECT 'X'
FROM PS_VAR_DATA_SPRG N, PS_COMMUNICATION O
WHERE N.COMMON_ID = B.EMPLID
    AND N.ACAD_CAREER = B.ACAD_CAREER
    AND N.STDNT_CAR_NBR = B.STDNT_CAR_NBR
    AND N.COMMON_ID = O.COMMON_ID
    AND N.VAR_DATA_SEQ = O.VAR_DATA_SEQ
    AND O.COMM_CONTEXT = A.SSR_SL_ACAD_YR_NLD
    AND O.SCC_LETTER_CD = '223')
    AND NOT EXISTS (SELECT 'X'
FROM PS_ITEM_LINE_SF P
WHERE P.COMMON_ID = B.EMPLID
    AND P.ITEM_TERM = '2' || SUBSTR( A.SSR_SL_ACAD_YR_NLD,3,2) || '0'
    AND P.LINE_ACTION = 'QCK')
    AND NOT EXISTS (SELECT 'X'
FROM PS_STDNT_EQUTN_VAR Q
WHERE Q.EMPLID = B.EMPLID
    AND Q.BILLING_CAREER = B.ACAD_CAREER
    AND Q.INSTITUTION = A.INSTITUTION
    AND Q.STRM = T.STRM
    AND Q.VARIABLE_CHAR2 LIKE 'PRE%')
    AND NOT EXISTS (SELECT 'X'
FROM PS_SSR_STD_PRG_NLD R
WHERE R.EFFDT =
      (SELECT MAX(R_ED.EFFDT) FROM PS_SSR_STD_PRG_NLD R_ED
        WHERE R.EMPLID = R_ED.EMPLID
          AND R.ACAD_CAREER = R_ED.ACAD_CAREER
          AND R.STDNT_CAR_NBR = R_ED.STDNT_CAR_NBR
          AND R_ED.EFFDT <= SYSDATE)
    AND R.EFFSEQ =
      (SELECT MAX(R_ES.EFFSEQ) FROM PS_SSR_STD_PRG_NLD R_ES
        WHERE R.EMPLID = R_ES.EMPLID
          AND R.ACAD_CAREER = R_ES.ACAD_CAREER
          AND R.STDNT_CAR_NBR = R_ES.STDNT_CAR_NBR
          AND R.EFFDT = R_ES.EFFDT)
    AND R.EMPLID = B.EMPLID
    AND R.ACAD_CAREER = B.ACAD_CAREER
    AND R.STDNT_CAR_NBR <> B.STDNT_CAR_NBR
    AND R.SSR_FORM_STUDY_NLD = 'E'
    AND R.ACAD_YEAR = A.SSR_SL_ACAD_YR_NLD
    AND R.EFFDT = (SELECT MAX( S.EFFDT)
FROM PS_SSR_STD_PRG_NLD S
WHERE S.EMPLID = R.EMPLID
    AND S.ACAD_CAREER = R.ACAD_CAREER
    AND S.STDNT_CAR_NBR = R.STDNT_CAR_NBR
    AND S.ACAD_YEAR = R.ACAD_YEAR))
    AND NOT EXISTS (SELECT 'X'
FROM PS_ACAD_PLAN U, PS_ACAD_SUBPLAN V, PS_ACAD_SUBPLN_TBL W
WHERE U.EFFDT =
      (SELECT MAX(U_ED.EFFDT) FROM PS_ACAD_PLAN U_ED
        WHERE U.EMPLID = U_ED.EMPLID
          AND U.ACAD_CAREER = U_ED.ACAD_CAREER
          AND U.STDNT_CAR_NBR = U_ED.STDNT_CAR_NBR)
    AND U.EFFSEQ =
      (SELECT MAX(U_ES.EFFSEQ) FROM PS_ACAD_PLAN U_ES
        WHERE U.EMPLID = U_ES.EMPLID
          AND U.ACAD_CAREER = U_ES.ACAD_CAREER
          AND U.STDNT_CAR_NBR = U_ES.STDNT_CAR_NBR
          AND U.EFFDT = U_ES.EFFDT)
    AND U.EMPLID = B.EMPLID
    AND U.ACAD_CAREER = B.ACAD_CAREER
    AND U.STDNT_CAR_NBR = B.STDNT_CAR_NBR
    AND U.EMPLID = V.EMPLID
    AND U.ACAD_CAREER = V.ACAD_CAREER
    AND U.STDNT_CAR_NBR = V.STDNT_CAR_NBR
    AND U.EFFSEQ = V.EFFSEQ
    AND U.ACAD_PLAN = V.ACAD_PLAN
    AND V.EFFDT =
      (SELECT MAX(V_ED.EFFDT) FROM PS_ACAD_SUBPLAN V_ED
        WHERE V.EMPLID = V_ED.EMPLID
          AND V.ACAD_CAREER = V_ED.ACAD_CAREER
          AND V.STDNT_CAR_NBR = V_ED.STDNT_CAR_NBR)
    AND V.EFFSEQ =
      (SELECT MAX(V_ES.EFFSEQ) FROM PS_ACAD_SUBPLAN V_ES
        WHERE V.EMPLID = V_ES.EMPLID
          AND V.ACAD_CAREER = V_ES.ACAD_CAREER
          AND V.STDNT_CAR_NBR = V_ES.STDNT_CAR_NBR
          AND V.EFFDT = V_ES.EFFDT)
    AND V.ACAD_PLAN = W.ACAD_PLAN
    AND V.ACAD_SUB_PLAN = W.ACAD_SUB_PLAN
    AND W.EFFDT =
      (SELECT MAX(W_ED.EFFDT) FROM PS_ACAD_SUBPLN_TBL W_ED
        WHERE W.INSTITUTION = W_ED.INSTITUTION
          AND W.ACAD_PLAN = W_ED.ACAD_PLAN
          AND W.ACAD_SUB_PLAN = W_ED.ACAD_SUB_PLAN
          AND W_ED.EFFDT <= TO_DATE('31-08-' || TO_CHAR( A.SSR_SL_ACAD_YR_NLD+1),'DD-MM-
YYYY'))
    AND W.EFF_STATUS = 'I'
    AND NOT EXISTS (SELECT 'X'
FROM PS_ACAD_PROG Y, PS_SSR_STD_PRG_NLD Z
WHERE Y.EMPLID = B.EMPLID
    AND Y.ACAD_CAREER = B.ACAD_CAREER
    AND Y.PROG_ACTION IN ('DISC','VDIS')
    AND Y.EMPLID = Z.EMPLID
    AND Y.ACAD_CAREER = Z.ACAD_CAREER
    AND Y.STDNT_CAR_NBR = Z.STDNT_CAR_NBR
    AND Z.SSR_PRG_ACTION_NLD IN ('MATR','RENR')
    AND Z.ACAD_YEAR = A.SSR_SL_ACAD_YR_NLD
    AND NOT EXISTS (SELECT 'X'
FROM PS_ACAD_PROG AA, PS_SSR_STD_PRG_NLD AB
WHERE AA.EFFDT =
      (SELECT MAX(AA_ED.EFFDT) FROM PS_ACAD_PROG AA_ED
        WHERE AA.EMPLID = AA_ED.EMPLID
          AND AA.ACAD_CAREER = AA_ED.ACAD_CAREER
          AND AA.STDNT_CAR_NBR = AA_ED.STDNT_CAR_NBR)
    AND AA.EFFSEQ =
      (SELECT MAX(AA_ES.EFFSEQ) FROM PS_ACAD_PROG AA_ES
        WHERE AA.EMPLID = AA_ES.EMPLID
          AND AA.ACAD_CAREER = AA_ES.ACAD_CAREER
          AND AA.STDNT_CAR_NBR = AA_ES.STDNT_CAR_NBR
          AND AA.EFFDT = AA_ES.EFFDT)
    AND AA.EMPLID = Y.EMPLID
    AND AA.ACAD_CAREER = Y.ACAD_CAREER
    AND AA.PROG_STATUS = 'AC'
    AND AA.EMPLID = AB.EMPLID
    AND AA.ACAD_CAREER = AB.ACAD_CAREER
    AND AA.STDNT_CAR_NBR = AB.STDNT_CAR_NBR
    AND AB.SSR_PRG_ACTION_NLD IN ('MATR','RENR')
    AND AB.ACAD_YEAR = A.SSR_SL_ACAD_YR_NLD))
    AND B.EMPLID = T.EMPLID
    AND B.ACAD_CAREER = T.ACAD_CAREER
    AND B.STDNT_CAR_NBR = T.STDNT_CAR_NBR
    AND T.INSTITUTION = B.INSTITUTION
    AND T.STRM = :1
    AND T.INSTITUTION = 'LEI01'
    AND T.EMPLID = X.EMPLID
    AND T.INSTITUTION = X.INSTITUTION
    AND T.STRM = X.STRM
    AND X.BILLING_CAREER = T.BILLING_CAREER)
ORDER BY 1, 2, 3
```

28 tables
10 subqueries

Months are calculated by subtracting startdates from end dates

Two-pass process for quitting and graduating

BACK TO THE DRAWING BOARD



Objectives:

1. Stop using just one query and create separate (small) building blocks
2. Get the student registration status per month in one overview

Advantages:

- ✓ Easier to oversee each block, and thus the complete process
- ✓ Easier to test every component of the batch
- ✓ Easier for troubleshooting
- ✓ Easier calculations by simplifying complex student registrations

SIMPLIFYING THE PROCESS FLOW (2017)

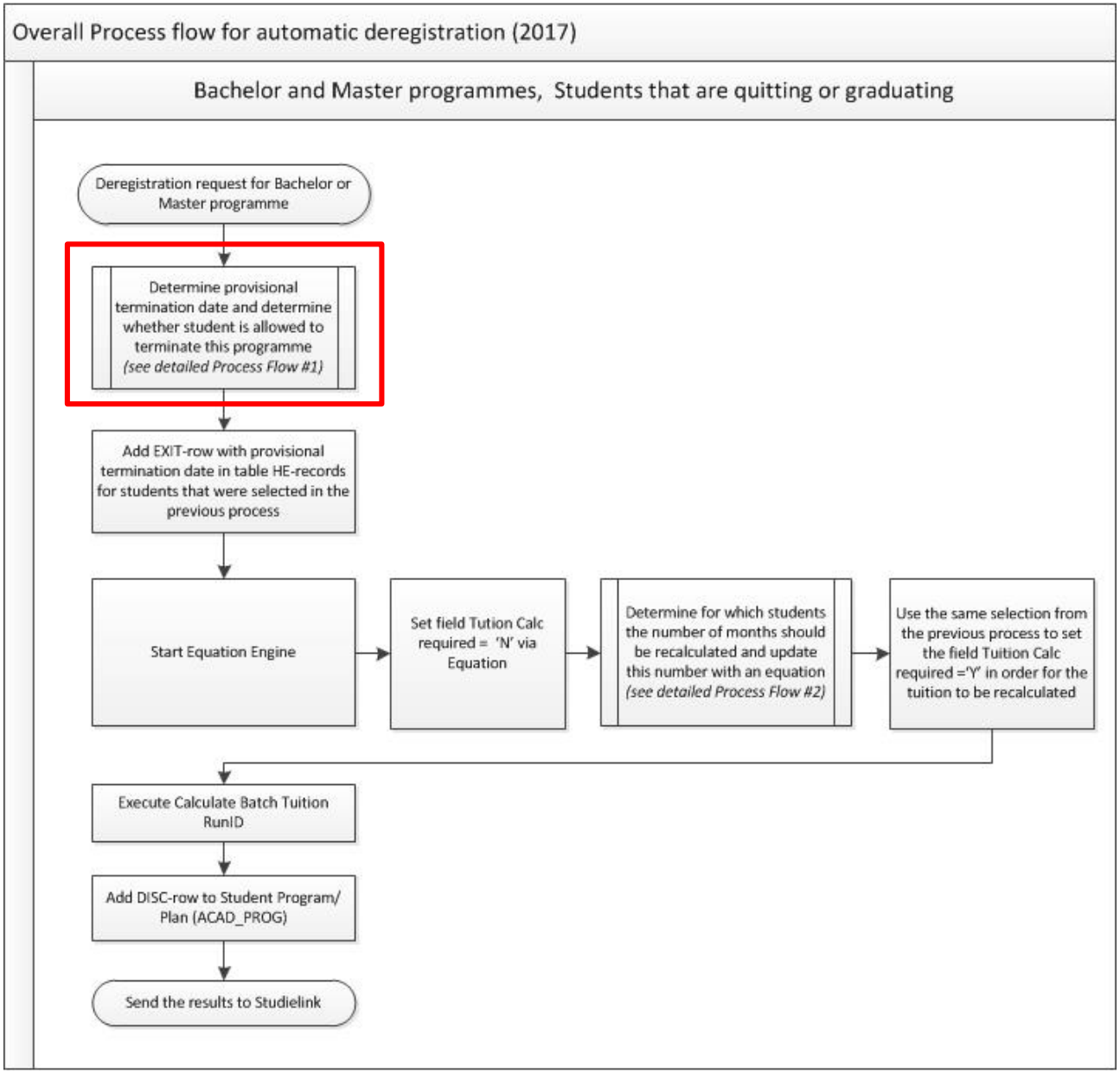
Identifying the **three** key elements of the process:

1. Determining provisional termination dates and determining whether the registration is eligible for automatic termination

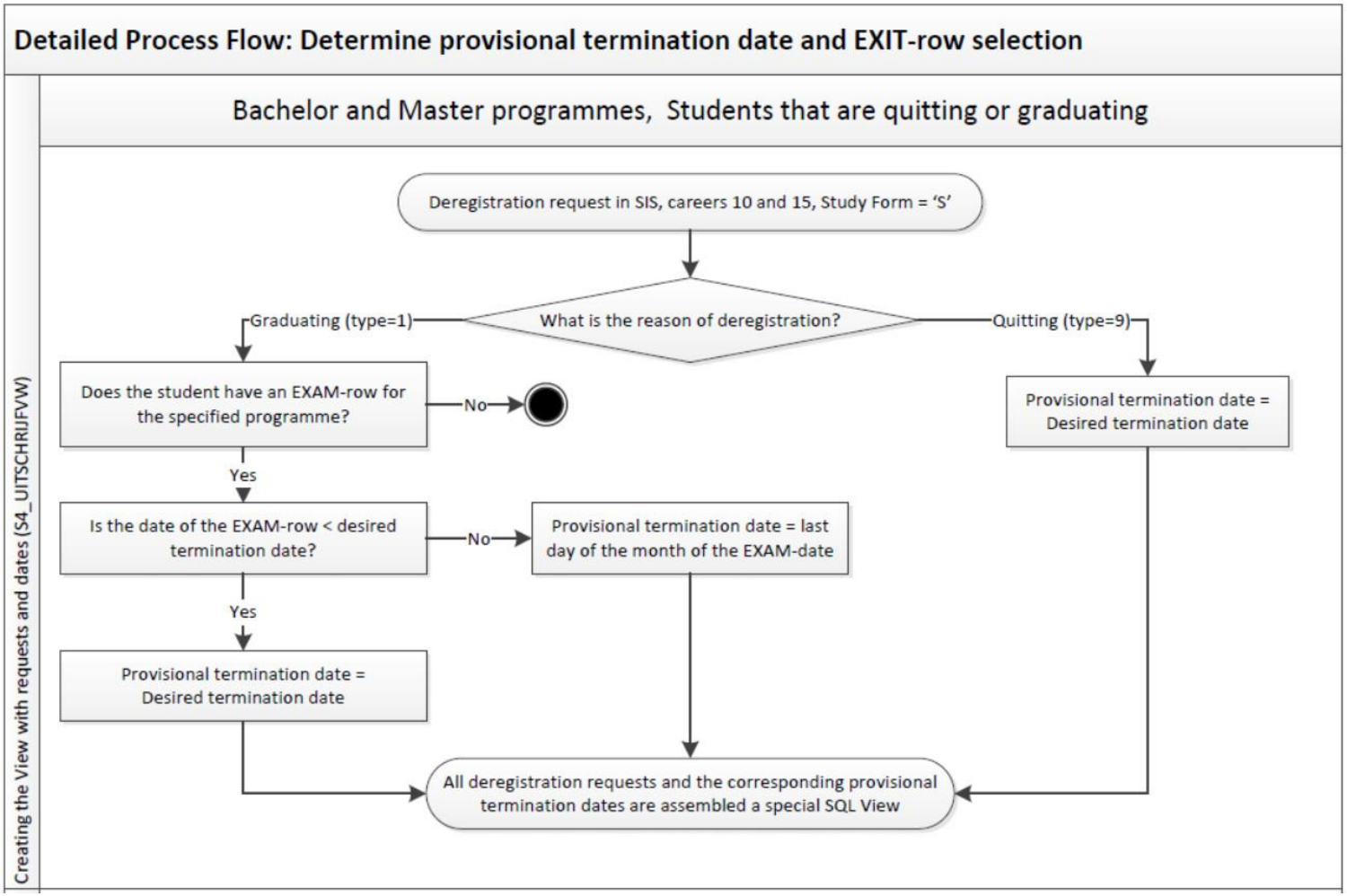
- Functional requirements
- Technical requirements

2. Determining whether the tuition needs to be recalculated

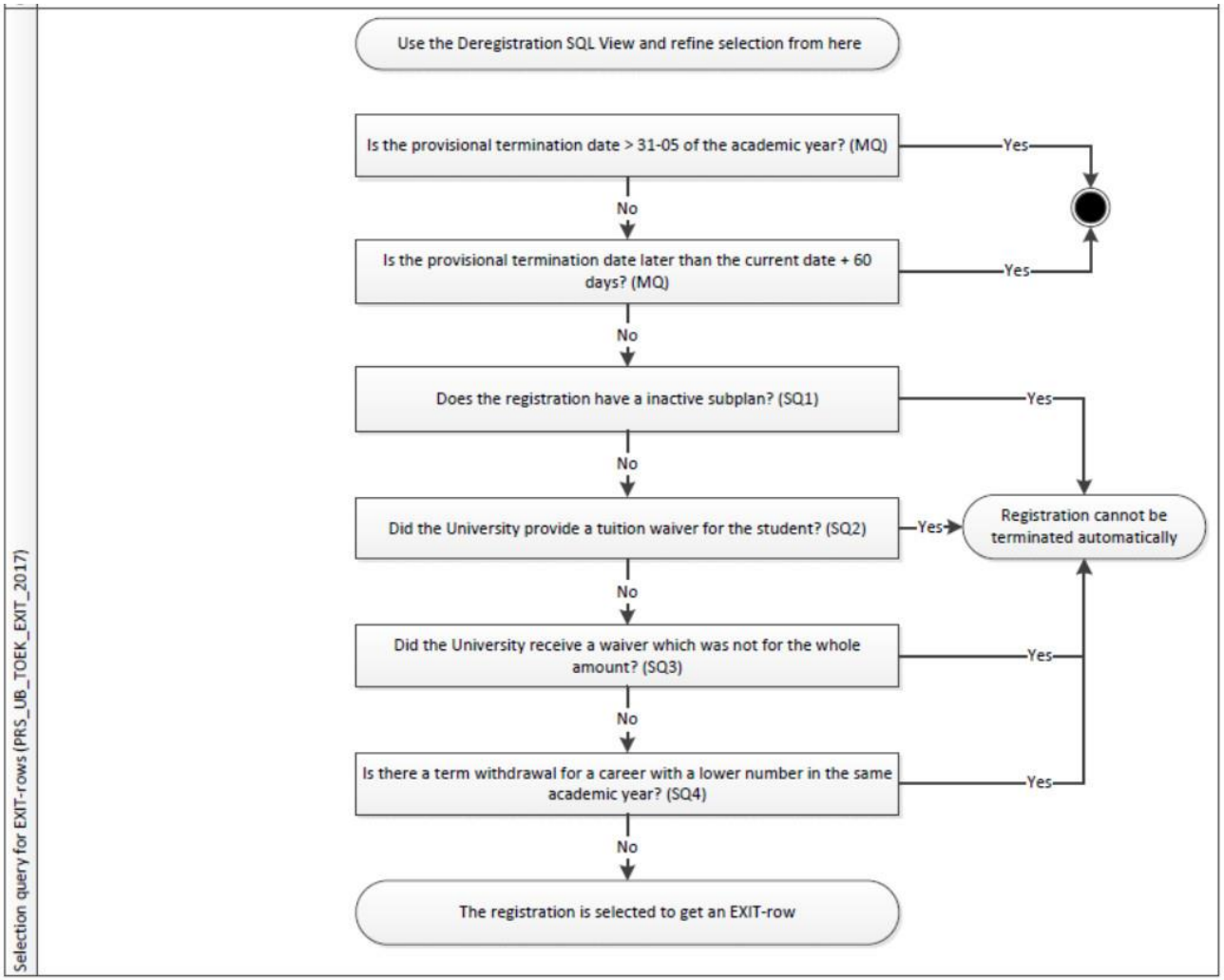
3. Calculating the correct number of registered months per student


Overall Process flow for automatic deregistration (2017)
Bachelor and Master programmes, Students that are quitting or graduating
Deregistration request for Bachelor or Master programme
Determine provisional termination date and determine whether student is allowed to terminate this programme (see detailed Process Flow #1)
Add EXIT-row with provisional termination date in table HE-records for students that were selected in the previous process
Start Equation Engine
Set field Tution Calc required = 'N' via Equation
Determine for which students the number of months should be recalculated and update this number with an equation (see detailed Process Flow #2)
Use the same selection from the previous process to set the field Tuition Calc required ='Y' in order for the tuition to be recalculated
Execute Calculate Batch Tuition RunID
Add DISC-row to Student Program/ Plan (ACAD_PROG)
Send the results to Studielink

UNIFY ALL DEREGISTRATION REQUESTS…



.. AND SELECT ALL ELIGIBLE REQUESTS



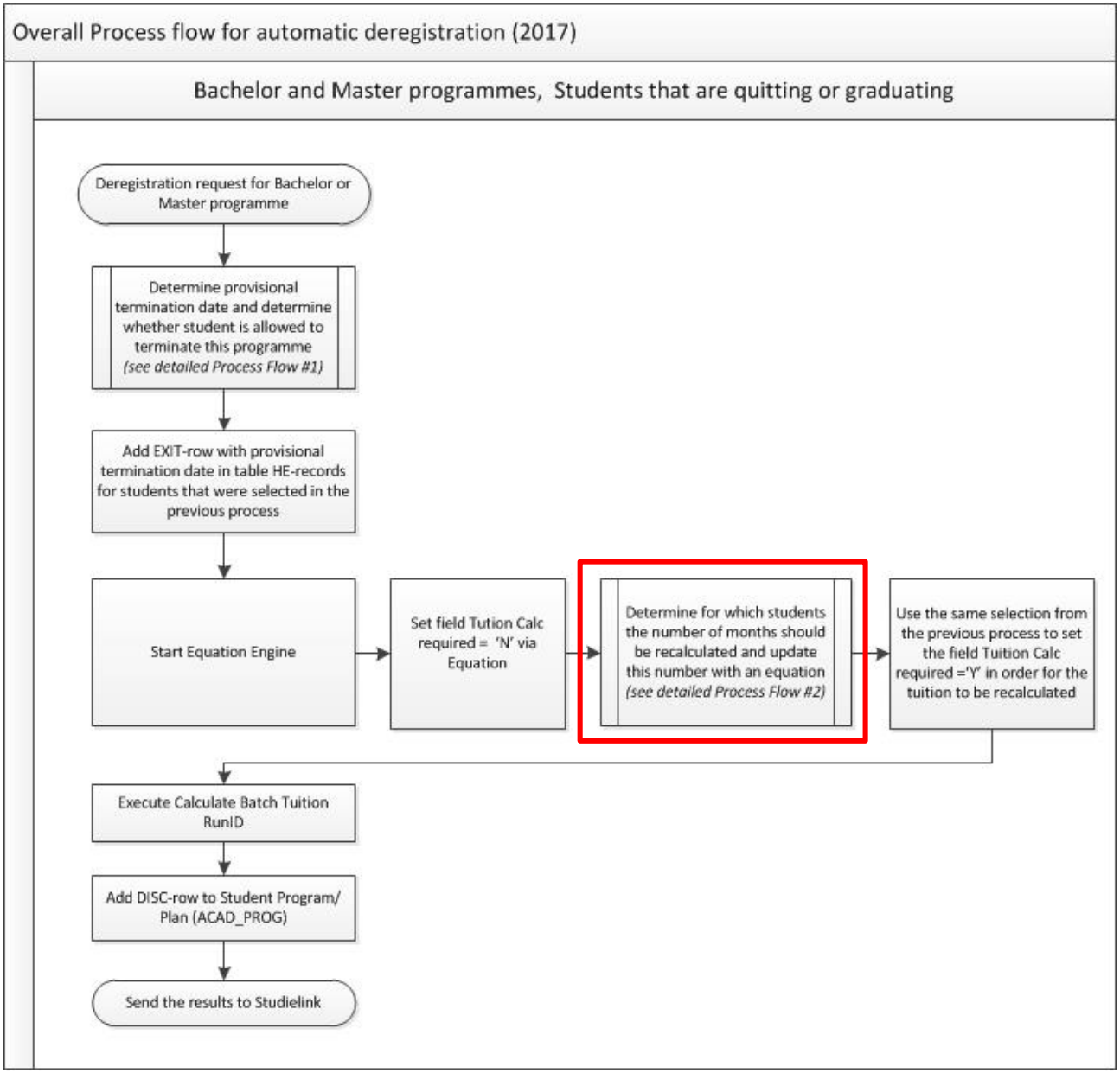
THE FIRST BUILDING BLOCKS

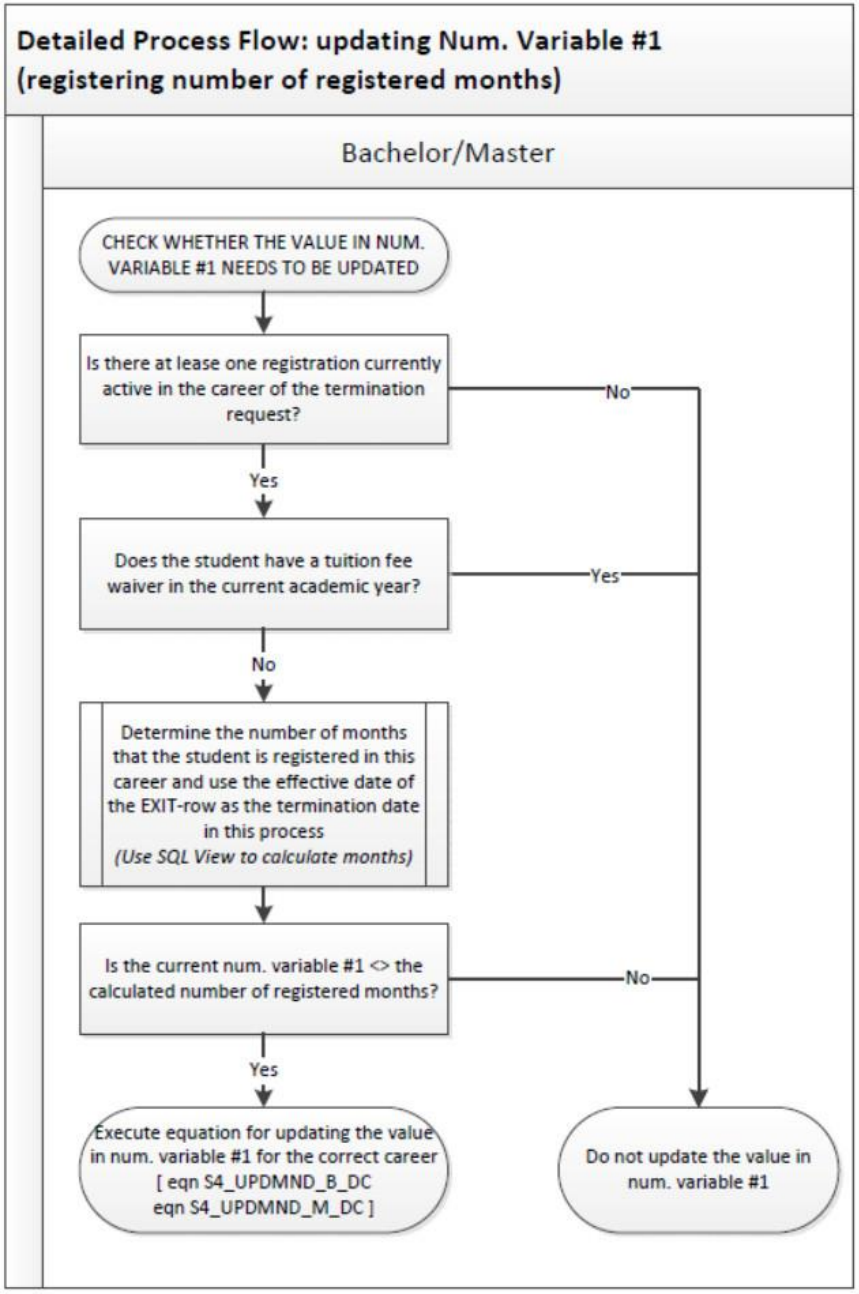
One SQL View containing all deregistration requests with the correct date

One query that allocates a new programme action called 'EXIT' which defines:

- whether the registration is eligible for automatic termination
- the provisional termination date

One query to monitor all exceptions that need individual attention (manual deregistration)


Overall Process flow for automatic deregistration (2017)
Bachelor and Master programmes, Students that are quitting or graduating
Deregistration request for Bachelor or Master programme
Determine provisional termination date and determine whether student is allowed to terminate this programme (see detailed Process Flow #1)
Add EXIT-row with provisional termination date in table HE-records for students that were selected in the previous process
Start Equation Engine
Set field Tution Calc required = 'N' via Equation
Determine for which students the number of months should be recalculated and update this number with an equation (see detailed Process Flow #2)
Use the same selection from the previous process to set the field Tuition Calc required ='Y' in order for the tuition to be recalculated
Execute Calculate Batch Tuition RunID
Add DISC-row to Student Program/ Plan (ACAD_PROG)
Send the results to Studielink



Determining whether an update of the variable is necessary.

If an update is necessary, we need to calculate the correct number of months.

CALCULATION WITH THE HELP OF SQL VIEWS

How to effectively use binary logic and aggregate functions in an SQL View to calculate how many months a student is registered

EMEA ALLIANCE 16-17 OCTOBER 2017

THE 2016 EXPRESSION

In 2016 we calculated the number of months with the following expression:

```
ROUND((TO_DATE(A.SSR_SL_END_DT_NLD) - TO_DATE(D.EFFDT)) /
(365/12))
```

This subtracts the start date from the termination date and then divides this by 365/12 and convert this to an integer using the ROUND function.

DISADVANTAGES OF THE 2016 EXPRESSION

It works for a student who is registered for one programme only

It will not work if a student is registered for more than one programme

If programmes do not have overlap it may also work (not always)

In 2016 we needed a different expression for graduating students

→ this was already fixed in 2017 by creating the SQL View with provisional termination dates

CREATING A NEW SQL VIEW USING BINARY LOGIC

A simple way to show student registration by using binary logic:

1 = registered / 0 = not registered

Student ID	Year	Career	Car Nbr	M09	M10	M11	M12	M01	M02	M03	M04	M05	M06	M07	M08
1234567	2017	10	0	1	1	1	0	0	0	0	0	0	0	0	0
1234567	2017	10	1	0	0	0	0	0	1	1	1	1	0	0	0

One expression for each month

Every expression checks the programme status for that month

CREATING A NEW SQL VIEW USING BINARY LOGIC

A simple way to show student registration by using binary logic:

1 = registered / 0 = not registered

Student ID	Year	Career	Car Nbr	M09	M10	M11	M12	M01	M02	M03	M04	M05	M06	M07	M08
1234567	2017	10	0	1	1	1	0	0	0	0	0	0	0	0	0
1234567	2017	10	1	0	0	0	0	0	1	1	1	1	0	0	0

By using the Maximum (MAX) function and deleting the field 'career number', we get the overview per career:

Student ID	Year	Career	M09	M10	M11	M12	M01	M02	M03	M04	M05	M06	M07	M08
1234567	2017	10	1	1	1	0	0	1	1	1	1	0	0	0

COMBINING BACHELOR AND MASTER DATA

Multiple careers generate multiple rows:

Student ID	Year	Career	M09	M10	M11	M12	M01	M02	M03	M04	M05	M06	M07	M08	Tuit. Type
1234567	2017	10	1	1	1	1	1	1	1	0	0	0	0	0	Stat. Fee
1234567	2017	15	0	0	0	0	0	1	1	1	1	1	1	1	Stat. fee

Using a second SQL View and some expressions can combine both rows into one:

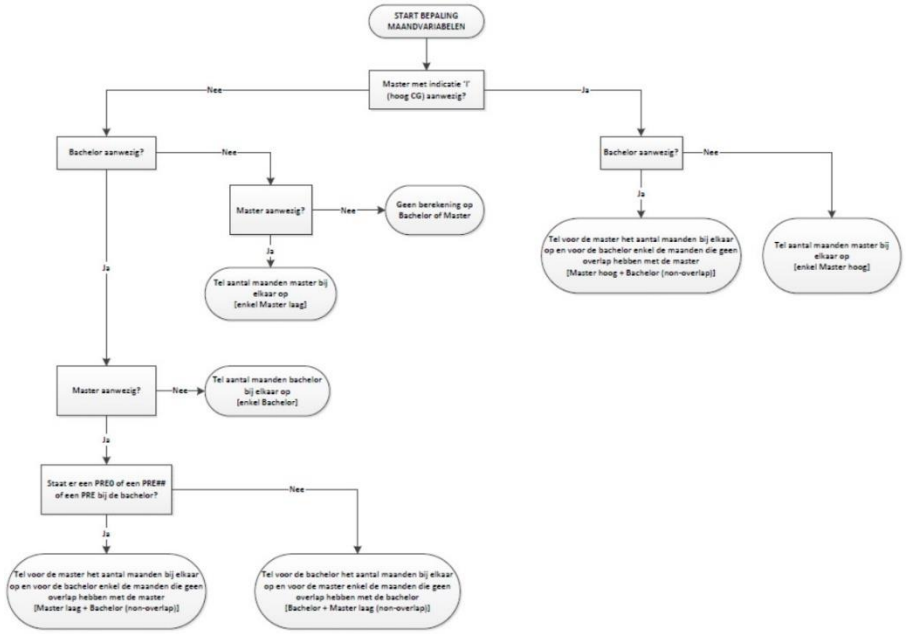
Student ID	Year	M09	M10	M11	M12	M01	M02	M03	M04	M05	M06	M07	M08	Tuit. Type	M09	M10	M11	M12	M01	M02	M03	M04	M05	M06	M07	M08	Tuit. Type
1234567	2017	1	1	1	1	1	0	0	0	0	0	0	0	Stat. Fee	0	0	0	0	0	1	1	1	1	1	1	1	Stat. Fee

The first sequence of binary fields is representing the Bachelor, the second sequence is representing the Master

COMBINING BACHELOR AND MASTER DATA

Combining the Bachelor and Master data has its own process flow, but we will not discuss this in detail.

Main rule:
Bachelor and Master add up to a maximum of 12 months per academic year.



COMBINING BACHELOR AND MASTER DATA

Again, the second view gives the following output:

Student ID	Year	M09	M10	M11	M12	M01	M02	M03	M04	M05	M06	M07	M08	Tuit. Type	M09	M10	M11	M12	M01	M02	M03	M04	M05	M06	M07	M08	Tuit. Type
1234567	2017	1	1	1	1	1	0	0	0	0	0	0	0	Stat. Fee	0	0	0	0	0	1	1	1	1	1	1	1	Stat. Fee

A third and last view counts the months per career and gives the output as follows:

Student ID	Year	Bachelor	Master
1234567	2017	5	7

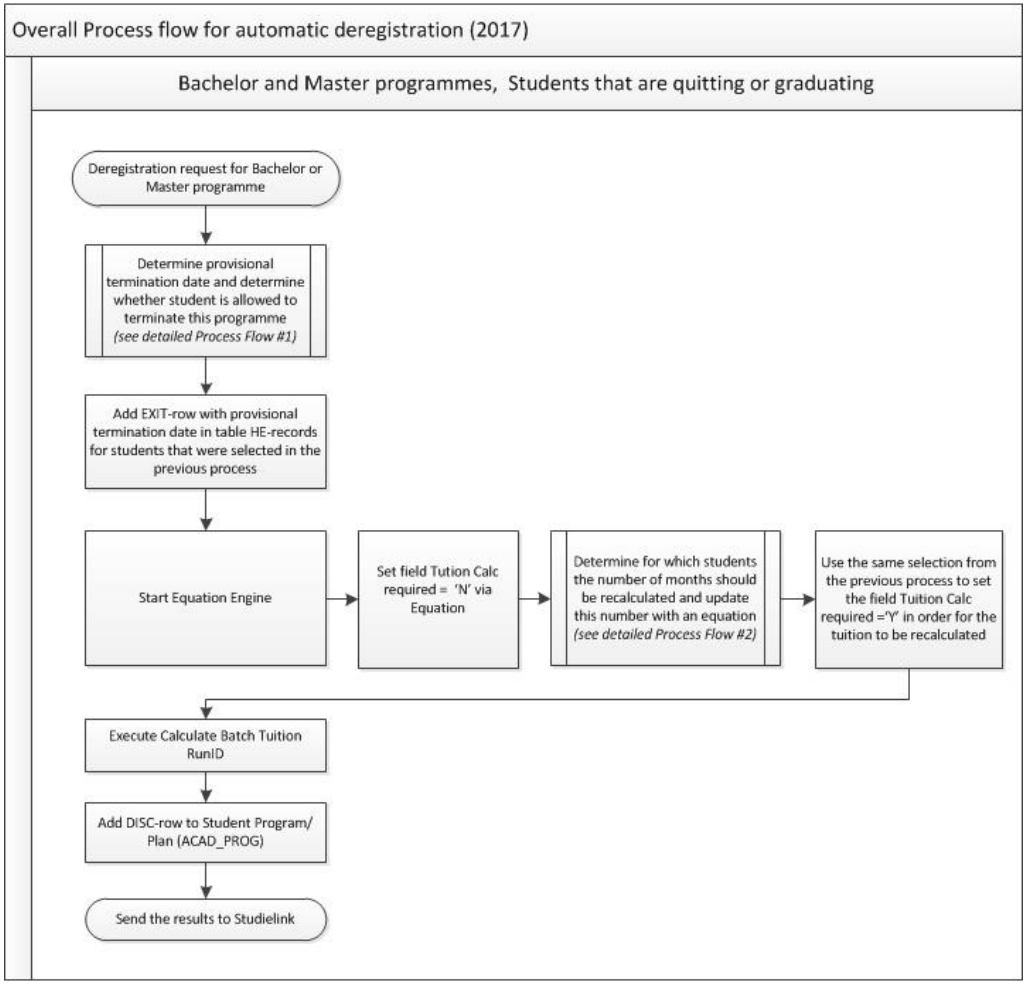
EQUATION ENGINE

We use the Equation Engine to run an Update statement to update the Num. Variable #1

Equation engine performs better than the Population Update Process (SACR > System Utilities) and it needs only one RunID.

Warning: no checks from Component Interface (!)

TRANSLATING THE 2017 PROCESS FLOW…



…TO A JOBSET DEFINITION

The whole process is now easily translated in a JobSet Definition containing 7 steps

Process	Description
Custom App Engine	Add EXIT rows to HE records
Equation Engine	Set TuitCalcReq= 'N'
Equation Engine	Update Num. Variable #1
SFPBCALC	Calculate Student Tuition (Batch)
Custom App Engine	Add DISC rows to Student Program/Plan
Custom App Engine	Delete EXIT rows from HE records
Studielink Process	Send messages to Studielink

EVALUATING THE BUSINESS CASE

- ✓ Batch runtime is avg. 9 minutes every night
- ✓ Waiting time for processing (for students) has drastically improved
- ✓ Every step of the process is visualized, which helps in easy Change Management and troubleshooting
- ✓ By dividing the process into building blocks, it is easy to troubleshoot when issues are reported
- ✓ No Peoplecode adaptations are necessary, everything can be managed by our own department

NOT THE ONLY WAY TO GO

This showcase is just an example of a possible route to follow, it is definitely not the only way

All sorts of complex processes can be divided into smaller pieces, for example by using SQL Views, or temporary data storing as a record in Student Program/Plan, in a Student Group, etc.

CONCLUDING THOUGHTS

ANY QUESTIONS?

PRESENTER

Hylke Hoekstra

Functional Application Manager

Leiden University

h.r.hoekstra@sea.leidenuniv.nl

Universiteit Leiden
The Netherlands

THANK YOU!

HEUG
HIGHER EDUCATION USER GROUP™