

# EUROCADCRETE

## AN IMPROVED DESIGN EXERCISE IN REINFORCED CONCRETE

**Ir. Hans (J.)C. Galjaard<sup>1</sup>, Prof.ir. Charles J. Vos<sup>2</sup>, Drs. Sabine Kunst<sup>3</sup>**

<sup>1)</sup> Assistant Prof., Delft University of Technology, Fac. of Civil Engineering and Geosciences

<sup>2)</sup> Prof., Delft University of Technology, Fac. of Civil Engineering and Geosciences

<sup>3)</sup> Educational technologist, Delft Univ. of Tech., Fac. of Technology, Policy and Management

*ABSTRACT: Presented are the results of 10 years of experience with a CAD/CAL reinforced concrete design exercise at Delft University of Technology. The exercise was developed in 1988 and 1989, implemented in 1989 for a test and in 1990 for regular use. Over 1200 students have used it since then. Students were asked to size and detail components of a simple reinforced concrete building, consisting out of columns, slabs and continuous beams at a workstation of the university CAD-Training Centre. The computer checked the results, gave comments on these results, and let the students correct them until found satisfactory.*

*Although the exercise was quite successful in the beginning, the success decreased in time because equipment and software got out-fashioned compared to other hard- and software students could use. Another drawback of the program was the very strict checking criteria used, which often tempted the students to solve the problem by 'trial and error'. This didactic unwanted situation was also reason to improve the program. In 1999 the workstations have been removed and the exercise could not be continued any more.*

*From several options available for the development of a new exercise, like upgrading the program or developing a complete new program, it was decided to adapt a commercially available program. In joint venture with a software consultant, Matrix Software BV, a complete new exercise is being developed, tested and implemented. The exercise is based on the existing commercial software from Matrix Software BV for the design of concrete structures. Another reason to select this program for the development of the exercise is that the structural analysis module is already being used at the university.*

*The program has several new features compared with the first one. It tries to implement some engineering judgement, by asking the student for answers based on rules of thumb, before computer-calculations are started. Furthermore the computer will not tell whether something is right or wrong according the code but it will show the implications of the result, leaving the judgement to the student. Another improvement will be that the exercise will contain some exercise in estimating and parameter-studies, asking for the effects of increase and decrease of sizes on the costs of a structure. The student can get help from the computer on different levels. Counting the amount and level of help being required and the time consumed may be used for a judgement. The program will use EuroCode 2 and will be made available for users who are interested all through Europe.*

**KEYWORDS:** *Computer Aided Learning, Concrete design, Computerised exercise, Concrete drawing, Structural Analysis*

### INTRODUCTION

It is a widely accepted opinion that civil engineers should at least have a basic knowledge of the design of concrete structures, and that a course in concrete structures should anyway form part of their curriculum. Furthermore it is generally believed that the students should not only have the theory explained to them, but for a better understanding also have to exercise, pref-



erably by designing and detailing components of a concrete structure. Such an exercise therefore formed part of the curriculum of civil engineering of Delft University of Technology for a long time. In the eighties however the (political) need was felt to reduce the costs and time of university education, which resulted in a major reorganisation including a drastic reduction of staff. An unchanged continuation of the former exercise was therefore not possible due to the large demand for staff to supervise this exercise. At that time computers and CAD applications were just becoming more easily available for general use, and after a preliminary study, it was decided to develop a new CAD-concrete exercise based on an (Auto)Cad environment. After an initial testing phase with a small number of students, the exercise was finally implemented in the regular curriculum for the academic year 1990/1991. Until the re-introduction of the fifth year in the curriculum in 1996/1997, the exercise was obligatory for all civil engineering students - about 200 per year. Due to re-arrangements in the curriculum after 1996/1997 the exercise is presently only required for about 120 students per year.

The so-called CAD-concrete exercise was implemented on HP-Apollo workstations, and was programmed in LISP and Pascal using AutoCad as a front-end. The students did not construct the concrete drawing themselves, but entered parameters in AutoCad from which a drawing was created. In 1998 the faculty decided to remove the workstations before 2000. For the continuation of the concrete exercise several alternatives have been investigated, like porting to a windows environment, switching back to a 'human' checked exercise, and the use of commercially available structural engineering software. A discussion with Matrix Software BV, the provider of the plane-frame structural analysis package used at the faculty of civil engineering, learned that they were enthusiastic about developing an exercise based on their existing software for concrete design. In 1999 it was therefore decided to develop the EuroCad-Crete exercise together with Matrix Software BV based on their commercially available software.

## **LEARNING OBJECTIVES**

For a better understanding it is important to know what skills the students will learn by doing the concrete exercise. After finishing the exercise:

- Students know the sequence of steps in which a concrete structure is being designed (i.e. the engineering process).
- Students know how to divide a complex (concrete) structure into 'simple' parts, for which they then can make a design in relation to the rest of the structure.
- Students are able to design individually simple (integrated) components of this concrete structure, like beams, columns and slabs. These designs will be checked with relevant codes. Special attention will be paid to the design and detailing of the bending and shear reinforcement of the components.
- Students are able to read and interpret a concrete drawing.
- Students have an insight in factors influencing the economy of the structure designed.

In order to achieve these learning objectives it is important to present the students a non-directive exercise which stimulates self-regulated learning. This implies a well-considered balance between the information and guidance offered by the computer and the expected input from the student. Special attention should be paid to kind of feedback students will get from the computer; unusable feedback will tempt the student to use a trial and error solving process.

## EXPERIENCES WITH THE 'OLD' CAD-CONCRETE EXERCISE

The 'old' CAD-Concrete exercise was based on the previous hand-checked exercise, and consisted out of the design of a concrete continuous beam, slab and column. Students got a general exercise, which could be individualised, based on their unique 'student-number'. Together with this exercise they got a detailed worked-out example of the exercise. The students were expected to make their personalised exercise with the help of design-tables and available sizing software. After this they could enter their data in the CAD-Concrete exercise program, and order the computer to check their design. The program had four distinct phases: In the first phase a check of the dimensions, and of some general data, like the cover, was carried out. It happened frequently that a student here already discovered that he had to redo all the calculations for his exercise, and often this was because one of the dimensions did not completely fulfil the requirements of the program. For instance: when the dimensions of the columns were not rounded to the nearest multiple of 25 mm. This kind of response by the program was found very annoying, in particular because they felt they had to redo a lot of work, while a simple warning would be sufficient for this kind of 'error'.

In the second phase the students had to detail all the reinforcement of the beam. For every group of bars the students had to enter the bar diameter and number of bars, and the beginning and end of the group. Furthermore they had to tell where the bars were positioned in the cross-section. For every section with constant shear reinforcement they had to enter the bar diameter, centre distance and number of bars. The computer then created a concrete drawing based on the data entered, see figure 1.

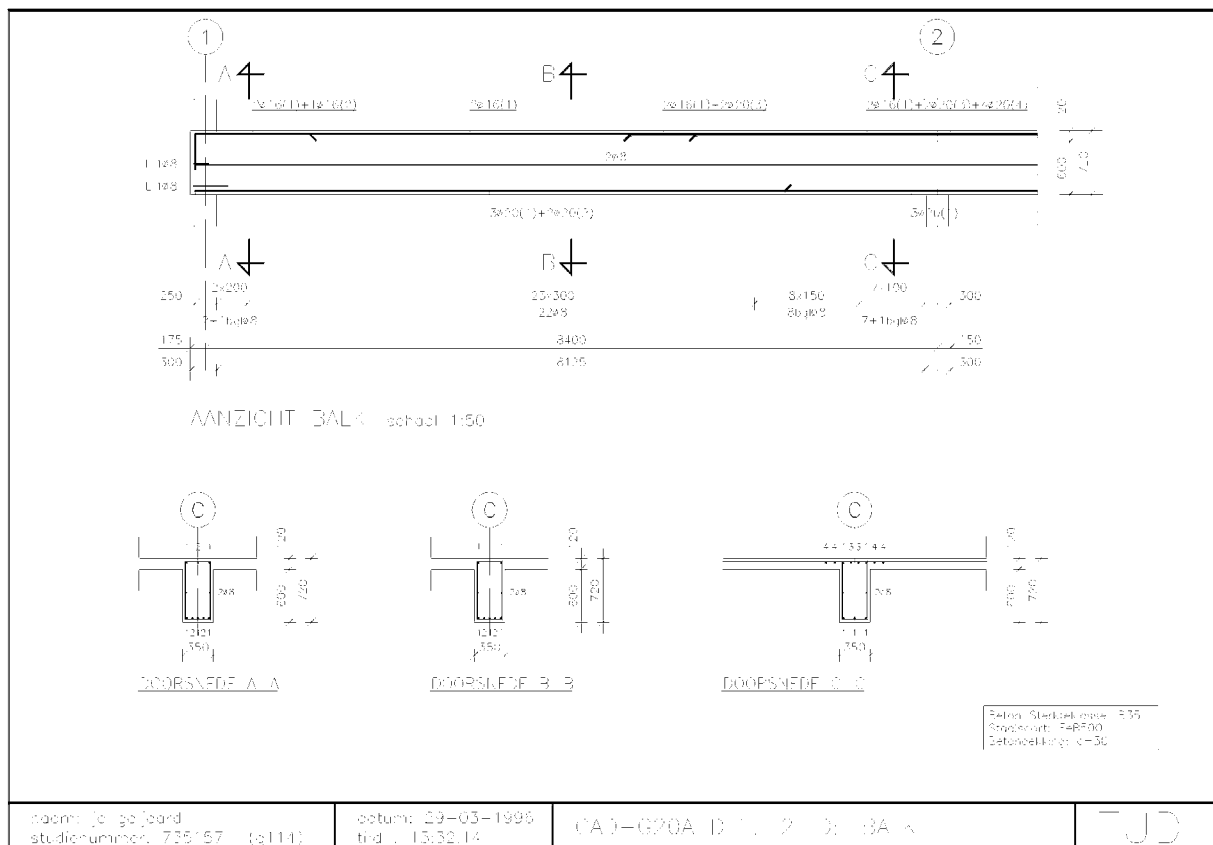


Figure 1: Completed concrete drawing for the beam of the CAD-Concrete exercise

Especially students with a higher student-number, and consequentially a more heavily loaded beam in an often more severe environment had large difficulties with this part of the exercise

due to the small deviations found acceptable by the program for the longitudinal and shear reinforcement. The program did not accept insufficient reinforcement, but also only accepted a little bit more reinforcement than required. The prior hand-calculation of those students anyway almost never fulfilled the demands of the program. Although the program gave a lot of feed-back in the form of tables and moment-cover diagrams the students hardly could use this information because they did not understand it, and they then tried to solve the given errors in the feed-back by trial and error. Sometimes the help text even guided them in the wrong direction, because the program indicated that they had to reduce the bar diameter for crack-width control, while in fact they just had to increase the amount of reinforcement in order to reduce the stress in this reinforcement. Even for experienced people it is was often difficult to find the correct solution with the help of the information provided.

The third phase consisted of the detailing of the floor slab. For every group of bars to be placed the students had to enter the bar diameter, and the position of the bars in the slab. From this data the computer created a concrete drawing, and checked the results. For this the program had to interpret the data provided. This was done by virtually assigning reinforcement to the various areas of the slab where reinforcement was required by the design code. This was followed by a check whether this reinforcement was in accordance with the design code, and generally whether bar distances matched adjoining fields. The interpretation could be misled by incorrectly or even un-placed reinforcement, which in turn would result in the wrong error message. The program could show how it interpreted the given reinforcement in the slab, but even with this help it was not always easy to locate the error. Students were sometimes trying to use trial and error to solve the errors given by the program, but frequently ended up in a loop. When shown how to interpret the error messages given they mostly quickly could finish this phase.

The last phase was a check of the column reinforcement. The students had to input the longitudinal reinforcement and stirrups, and had to show how the longitudinal reinforcement was distributed over the cross-section. Apart from incorrectly read error messages the students could generally speaking finish this phase within a quarter of an hour.

From the previous sections the impression might be obtained that the program had many problems what certainly was not the case. Contrary, the program has worked very well during the last decade, and more than 1200 students successfully finished the exercise. At the same time the amount of work involved of the staff to supervise this exercise has been minimised. Due to the leave of the developers of this piece of software it even had to run without significant maintenance, what might be regarded as a miracle.

## **DEMAND FOR CHANGE**

A well known saying tells: "Never change a working program", so why change the CAD-Concrete exercise? The most important reason was that the hardware, HP-Apollo workstations, would be switched off before the Y2K because they were too old to be made millennium-proof. Parts of the CAD-Concrete exercise were moreover specifically coded for this particular hardware, which made is less easy to simply port it.

There are however more good reasons:

- Although the program was very modern at the time when it was first used, it now had become rather old-fashioned compared to the other much more fancy programs students can use at our faculty. At the start student were quite enthusiastic and impressed by the program, but in time this feeling decreased to a necessary old-fashioned exercise they were obliged to do for their study.
- Programs nowadays preferably run in a WINDOWS environment, and this was preferred for the 'new' exercise too. This however made a simple port less likely.

- The original developers of the software did not work for Delft University of Technology anymore. Furthermore the software was not documented very well, the only documentation in fact were comments written in the source code.
- New codes, like EuroCode 2, are coming, and the exercise should be updated to these new codes. This again would not be too easy, see the previous point.
- There were questions about the teaching/learning effect of the exercise. In order to quickly pilot the students through the exercise the worked-out example that they received on forehand had to be quite detailed. If not, then the student had needed much more time to complete the exercise, and they would mainly have used 'trial and error'. Now however most students copied the example with their own numbers probably without understanding what they were doing. Some students even only put their numbers in the copy of the worked-out example they had received.

### **OPTIONS FOR A NEW EXERCISE**

For the 'new' exercise several options were investigated into more or less detail. These options were:

- Porting the old CAD-Concrete exercise to a WINDOWS environment.
- Developing a complete 'new' exercise.
- Switching back to a hand-checked exercise again.
- Developing a program which could read an interpret 'real' drawings delivered by the students, which would be based on neural network technology.
- The use of professional software for structural engineers.

By porting the old CAD-Concrete exercise to a window environment a number of aforementioned problems would not have changed, like the inadequately documented program, the need to update it to EuroCode 2 in the near future and other maintenance problems, and most importantly the teaching/learning effect of the exercise.

The development of the 'old' CAD-Concrete exercise had been possible because of a grant of the TU Delft, and costed more than two man-years. A new grant was not very likely, what made this option unaffordable.

A switch back to a hand-checked exercise would have been possible with the current rather small number of students. Due to some developments in the faculty it is however not too unlikely that this exercise will be part of the base course of all civil engineering students in the future again, and the number of students expected to do the exercise will be too large to handle by the current staff.

The development of a program using pattern recognition techniques, based on neural network technology, which could evaluate the drawings delivered by the students would even cost more than the development of a 'new' CAD-Concrete exercise. Furthermore it would be likely that the development of such a program would cost a lot of time, and that it would not be ready in April 2000.

The only option seemed to be the development of an exercise based on professional software for structural engineering. The first idea was that students would make the exercise by hand, discuss the results with one of the supervisors, and if roughly found satisfactory make a comparative design with a professional concrete calculation program. After this they could compare this calculation with their hand-calculation, and judge how realistic their own design was. A discussion with Matrix Software BV learned that they saw other opportunities, like using their software to check the student's data of their hand-calculation. Their basic idea was to use existing modulus of their concrete designing/checking program for the development of a wizard like program which would step by step check the students design activities. In September 1999 it was agreed that Matrix Software BV and the TU Delft together would develop

such a program for the new exercise called EuroCadCrete. The development is divided into three steps: first the development of the basic exercise based on the current Dutch concrete code. Secondly the development of a parameter study to show the effects of variations in the designs on the economy, and thirdly to implement the rules for EuroCode 2.

## OBJECTIVES OF THE NEW EUROCADCRETE EXERCISE

The objectives for the new EuroCadCrete exercise have also been based on the experiences with the old CAD-Concrete exercise. The objectives are:

- Students should design a concrete beam according to the relevant codes, which design should be checked and corrected without human interference of supervisors.
- The design should leave some 'engineering judgement' to the student. More specific: checks which could have been included to prevent uneconomic designs should as much as possible be avoided.
- The judgement on the economy of the design compared to other possible designs should be left to the student by showing him a cost comparison of some alternatives.
- Solving the exercise by trial and error should be prevented by giving the student the possibility to get context related help, varying from a hint to the correct answer. Asking too much help should however be discouraged.
- The students should size the structure by global rules of thumb before doing any detailed calculations.
- At the start of the exercise the students should determine the correct sequence of the steps of the design process.
- The student will only be allowed to proceed to the next step of the design process when the current step is fully correct. In this way it will be immediately clear where errors are made, and misleading error messages which are the result of the subsequent stacking of errors in a number of steps can be avoided.

These objectives have led to a schedule comprising all the necessary steps of the exercise. In this schedule a differentiation is made between:

- a the task to be done
- b the product of this task
- c the required activity of the student for this task
- d the required activity of the computer for this task
- e learning objective of this task

This schedule is presented in table 1 and formed a starting point for the development of the EuroCadCrete exercise.

*Table 1: Schedule for the EuroCadCrete exercise*

	<b>A Task</b>	<b>B Product</b>	<b>C Students activity</b>	<b>D Computer activity</b>	<b>E Learning objective</b>
1	Fixing preliminary static scheme, from a drawing with views of a tunnel-roof or a continuous beam in a prefab building	Static scheme with sizes and loads and other design parameters.	Making sketches, static scheme, loading scheme with EC 1 and common sense, and environmental class.	Only administrative generating of a unique set of design parameters.	Training the relation in between the constructed world and a structural scheme for analysis.
2	Selection geometry of structural concrete and final static scheme.	Shape (including possible haunches) and depth, width, cover and concrete quality of the cross section.	Simple trial & error hand analysis, for moment and shear capacity of cross section. Prepare input.	Fixes data from student. Analyses and shows range of valid parameters. Depth for min. and max. $\zeta$ .	To analyze by hand shape, sizes and concrete specifications from a beam type concrete member.
3	Analysis from ruling external and internal forces	Ruling bending moment and shear diagrams	Analysis by Frame program on PC	Automatic analysis and showing results to student for comparison.	Understanding from force-distribution in a structure. Using a frame program in practice.

	<b>A Task</b>	<b>B Product</b>	<b>C Students activity</b>	<b>D Computer activity</b>	<b>E Learning objective</b>
4	Check ruling cross sections on moment and shear for check on SLS and ULS	Reinforcement % longitudinal reinforcement and stirrups in ruling cross sections	Analyzes by table (or simple PC program) with intermediate steps (depth compression zone etc.) the reinforcement. Checks for crack-width.	Analyzes and shows intermediate steps after student has provided his intermediate data.	Getting acquainted step by step with the complete process of analyzing a cross section and the influence of the relevant parameters.
5	Designing a practical and economic reinforcement	Full drawing of a rational reinforcement as far as structurally required	Input of reinforcement cover diagram, choice of diameters, basic reinforcement and secondary bars.	Automatic analyzing and showing of reinforcement cover diagram, commenting students diagram (You are using x % to much or ...) Analyzing students diagram on consequences. Just the same for stirrups, c.t.c. distances of bars and diameters.	Exercising steps required to arrive at a rational reinforcement from a cross section analysis only.
6	Find reinforcement details	Complete reinforcement drawing	Indicate all other required reinforcement	Check per component being used as input and comment deviations from rules with consequences. Generate and comment answers on some interactive questions on detailing.	Getting a complete picture of detailing reinforcement.
7	Cost estimate	Estimate, being build up from unit prices.	Estimate unit costs per cum of concrete for the whole structure. (Concrete, formwork and reinforcement) Use these quantities as input and give the present the price.	Analyze and show the exact price of the structure after the student has given his figures.	Obtaining an idea on the cost structure of a concrete construction.
8	Parameter-study	Influence of important design parameters on the costs of a concrete construction	What is the price difference for a 50% higher loading, or when paint is being used, that bridges cracks.	Analyzes and shows these answers exact after student has presented his estimates. Presents a way of improved guessing, when required.	Obtaining an idea of the influence of different parameters on the costs of a concrete structure

## IMPLEMENTATION

The implementation of EuroCadCrete is based on existing software of Matrix Software BV, and in particular on Matrix Frame and their existing code checking routines for the Dutch concrete code<sup>1</sup>. The student is supposed to enter the geometry of his structure, the loads and the loading-combinations in MatrixFrame. Eventually this part of the exercise could be done at his own home-PC using the limited student version of MatrixFrame. For this part the student needs to estimate the sizes of the structural components used in his frame by rules of thumb.

The student can obtain information regarding his exercise from a manual, which comprises of the necessary data to construct his own personalised exercise, and some basic information for the calculation and distribution of the reinforcement in the beam. Furthermore some necessary information is given for data required by EuroCadCrete.

With MatrixFrame the student can calculate all governing moments and shearforces in the ultimate- and service limit-state. Based on these results he can make all the necessary calculations for the input in EuroCadCrete, and take these results to the computers where EuroCad-

<sup>1</sup> In the near future EuroCode 2 will be used instead

Crete is installed. He now can either input his data for the geometry, loads and loading-combinations in the computer by hand, or let MatrixFrame read the diskette with his own data developed at home. When all the forces are calculated by MatrixFrame he can start EuroCad-Crete by clicking the appropriate button. EuroCadCrete starts with a check of the geometry, concrete strength class, loads and loading-combinations. An error message will be given if the data is not in accordance with the student's personal number. If ok, than EuroCadCrete really starts, and the student is first asked for some general information, like structural member and structure type (beam, slab spanning in one direction, strip), steel name, way of fabrication, nominal aggregate size and the size of the vibrator used.

In the next step, see figure 2, the student has to determine the correct sequence of the design steps. To make it somewhat easier the process has already been divided into four blocks, of which the first and last block are given. Above this the other steps have already been grouped in two blocks, one for the cross-section related steps and the other for the global reinforcement layout steps. For every block the students can choose steps from a list which only shows steps not used in previous steps.

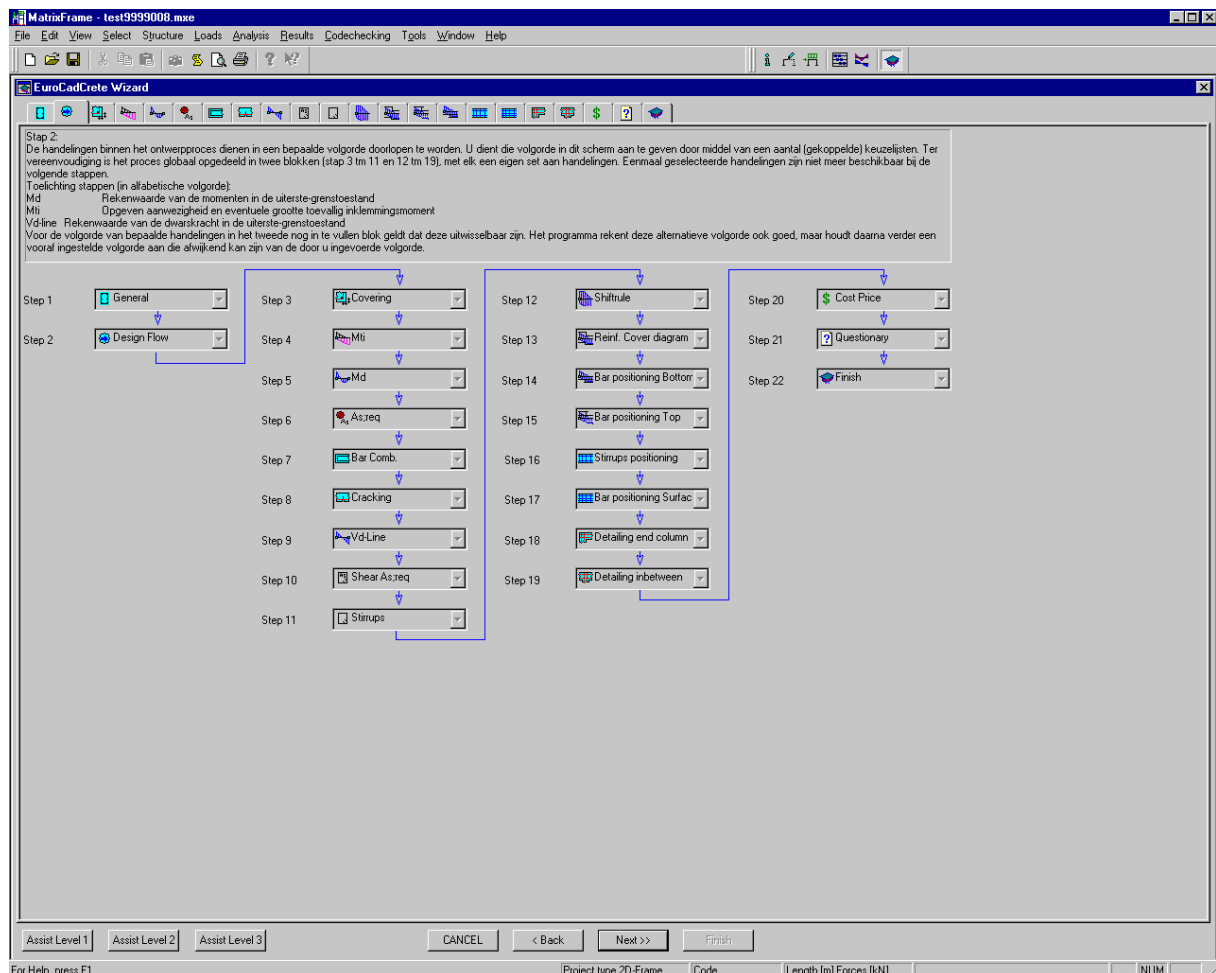


Figure 2: Sequence of design steps

Figure 2 also reveals the layout of the screens. On top the tabs off all steps of the program can be seen, indicating the progress. These tabs become visible when the second step is successfully finished. Every screen will show some explanatory text, and most screens also show a graph with information specific for that screen. On the bottom three assist buttons can be seen for help on three different levels. The first help level or Assist 1 is free and shows some com-



common information related to the active cell. The second help level or Assist 2 has a penalty of one point, and shows more specific information for the active cell. The third help level with a penalty of two points gives the correct answer for the active cell. It is not possible to get help on level 2 without asking for help on level 1, and on level 3 without help on level 2. The <next> button becomes available when everything on the screen is found all right. The value of help in points will be adjusted by experience.

For the first phase of the development the program is divided in 4 blocks with 22 steps:

I For block 1:

- 1 general information
- 2 sequence of design steps

II For block 2:

- 3 environment and cover
- 4 type and size of supports, implied moments
- 5 governing sections and moments
- 6 required cross section of bending reinforcement for ultimate limit state
- 7 possible bar combinations, split in basis reinforcement and additional bars
- 8 check for crack width, could be alright, otherwise new bar combinations to be specified
- 9 governing sections for shear force
- 10 required cross section of shear reinforcement
- 11 stirrups to be provided, basic basket and additional stirrups

III For block 3:

- 12 shift of moment diagram
- 13 reinforcement cover diagram, entering the beginning and end of the active length of a rebar, entering the anchorage length, see figure 3
- 14 bar position bottom, entering begin and end of the physical rebar,
- 15 bar position top, entering begin and end of the physical rebar
- 16 stirrups positioning, basic basket, number of additional stirrups
- 17 positioning surface reinforcement
- 18 detailing end column connection, multiple-choice question
- 19 detailing intermediate column connection, multiple-choice question

The sequence of the steps 14 and 15, 16 and 17, and 18 and 19 are equivalent and hence can be exchanged.

IV For block 4

- 20 calculation of cost price
- 21 questionnaire, opinion about program
- 22 results, total of penalty points, time used, number of assist requests

In the future a new step will be added between step 20 and 21 to show the students the effects of variations in the design on the economy.

When the total penalty at the end of the exercise is excessive, the student will have to do new (other) exercise.

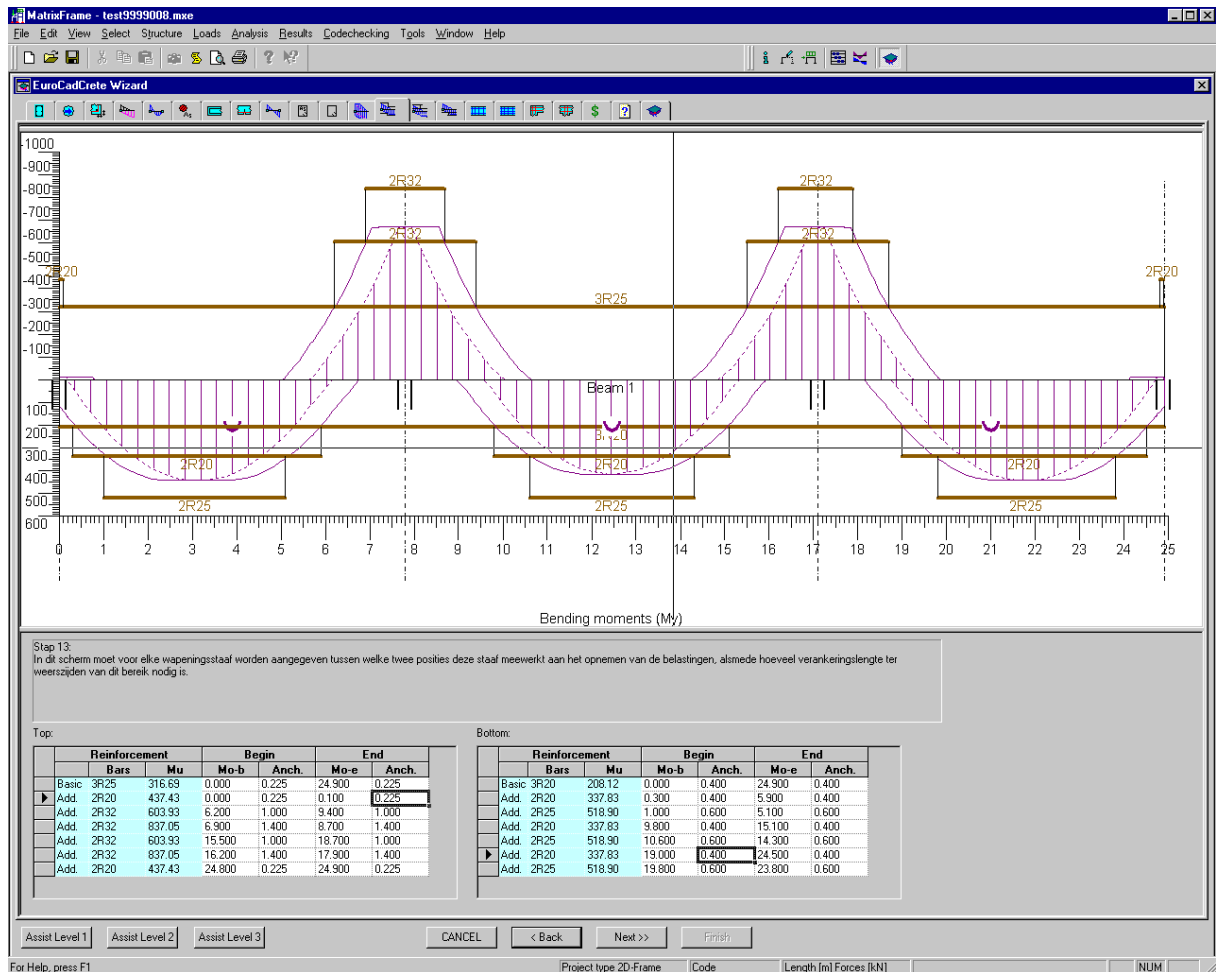


Figure 3: Step 13, reinforcement cover diagram

## FIRST EXPERIENCES

Unfortunately at the date of submission of this paper only a few experiences have been collected due to unforeseeable delays in the installation of the  $\beta$  testing version. Ten students who were invited to test the program have hardly started. A few of them, who also did the previous CAD-Concrete exercise, remarked that this exercise felt much more 'friendly'. Some of the staff who have tested the program directly felt accustomed with it.

We will provide the latest information during the conference, at which moment we expect that approximately 100 students will have finished the exercise. A complimentary hand-out to this paper will be made available.