ICT – Competencies in Mathematics and Informatics

Hans-Stefan Siller / Karl Josef Fuchs

Department of Mathematics and Computer Science Education, University of Salzburg, Salzburg, Austria

The paper mainly bases in the belief that Mathematics and Informatics contents have been taught together for many years. Hence Mathematics' and Informatics' competences' models have shared well accepted techniques and strategies for many years. Modelling in its different subject – specific requests has emerged as pivotal idea in all these concepts. The presentation generally focuses on the aspect of Functional Modelling covering application – and special implementation abilities through the use of the Computer Algebra System (CAS) Mathematica^{TM}.

The usage of this CAS is founded in its functional design and working. With this software students have to think about functional aspects for implementing algorithms so that geometrical aspects can be explored. Because of the cross-disciplinary aspect this software has one more big advantage: the elementary implementation of imperative-procedural modules. With their help it is possible to write a code that returns an adequate output for students.

Based on the central idea of modelling a schedule (describing the problem/mathematizing – graphical representation (with the help of PROGRAPH diagrams) – coding – testing) should be followed, so that students can understand the complex structures sustainable. With the help of a graphical representation as well for the functional part as for the imperative-procedural part students recognize the difference between the different programming paradigms in computer-science. Following this representation students have to implement and experiment with the implemented code so that the geometrical aspects get obvious.

In the paper the importance of (functional) modelling is represented by actual didactical literature in Informatics and Mathematics because this topic is gaining in importance.

Keywords: Competences, Functional Modelling, Innermathematical Problem

1. Preface

Teaching mathematical contents in Informatics has a long tradition. An even longer tradition is to teach informatical contents in Mathematics. This will not be very astonishing if we have a closer look at the history of both subjects. When the usage of computers got common a lot of mathematicians were impressed by the possibilities such machines offered. Because of this enthusiasm they tried to implement mathematical algorithms and calculations with the help of different systems to computers. The triumphal procession of computers in mathematics and mathematics education started and has lasted until today. Other important aspects which have strongly gained influence on teaching and education have been the different activities in the development of national educational standards in Mathematics and Informatics.

1.1 Standards in Mathematics in Austria

Right now they are developed and implemented for Mathematics in Austria based on a model of competence in Mathematics [6] which has been developed by a ministerial group of experts. The different cells which have been filled and will still be substantiated by *prototypical* tasks can be illustrated by a cube.

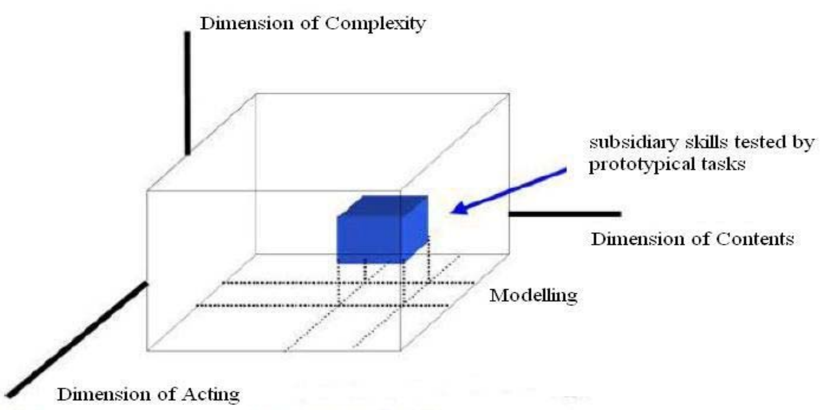


Fig. 1 Austrian model of competences for Mathematics [11]

Acting, contents and complexity are the three dimensions. *Modelling* is an important item in the dimension of acting.

1.2 Standards in Informatics in Austria

The very different situation in Informatics that has been strongly affected by the very diverse role of Mathematics on one hand and Informatics on the other hand in school in Austria is very unsatisfying actually [2]. Suggestions of Informatics educators [3] with little influence on teaching in school are complemented by special ministerial working groups addressing vocational education. The model of competences they have developed is two – dimensional.

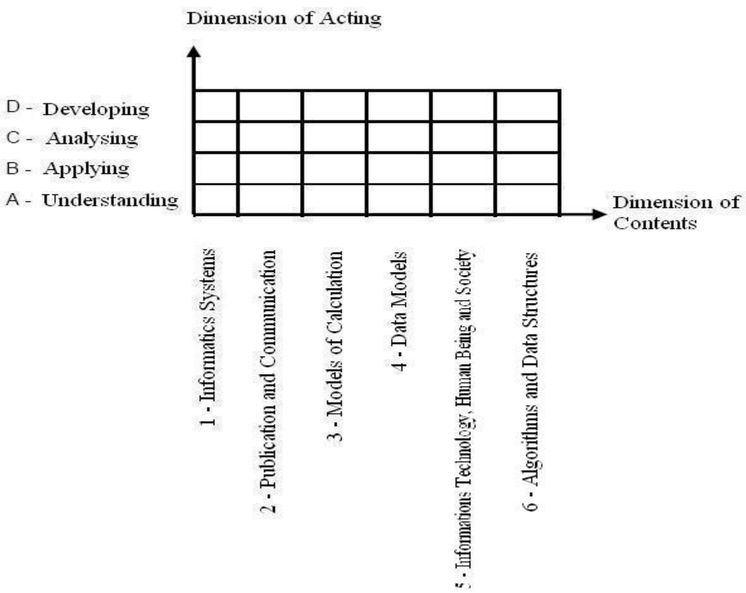


Fig. 2 Austrian model of competences for Applied Informatics [1]

Most of our students do not want to reflect about the outcomes of their calculations; they are happy about certain result and do not think about interpreting or drawing some conclusions. Therefore it is necessary to encourage modelling aspects in education and to show how "easy" modelling can be done in Mathematics or Informatics.

2. Functional Modelling

If certain problems are realized *functions* or *functional dependencies* will be necessary to map the behaviour of the investigated situation. This is the –let's say - mathematical side of modelling. Switching over to the informatical point of view puts different programming paradigms [4] in perspective. Wading through the different concepts of programming we will recognize that there exists a style to implement *functions* or *functional dependencies* in an elementary way. We decided to take account of the Computer Algebra System (CAS) Mathematica^{TM} which is dominantly used in Mathematics. Arguments for the suitability of this CAS are presented in [7, 9, 10].
A draft of a *modelling* process will be shown in the following part.

Discussion of the problem

Since certain topics like quadratic equations, highest common factor/lowest common multiple are discussed that often we want to focus on a side issue in Informatics – the topic of Geometry. The advantage of this topic is the opportunity to study the influence of parameters with the help of a simple functional request.
One could say that this topic is too inner-mathematical and therefore not suitable for modelling. In our opinion this argument does not hold because students can see that algebraic equations, like the equation for a circle or an ellipse, can be treated as (implicit) functions.
Another important argument is the combination of Analysis and Geometry which is not discussed deep enough in education. For this request a simple modelling process could be started in Informatics. The students should construct a little program that is able to calculate the relative positions of two or more conic sections.

Let us have a look at such an example:
Given the circle k: $(x-x_{k0})^2+(y-y_{k0})^2=r^2$ and the ellipse ell: $b^2\,(x-x_{e0})^2+a^2\,(y-y_{e0})^2=a^2\,b^2$.

- Find the intersection points of both curves and calculate the solutions of both equations.

The techniques to solve the problem without technology are very easy. But if we aim for generalizing the given problem a typical *modelling* process in Informatics will be started.

Step #1: Summing up the given problem

A circle and an ellipse in general description are given. For calculating the relative position of both geometrical objects Function modules should be constructed. The calculation should put out all non complex solutions of the both equations.

Step #2: Building a graphical representation

A graphical representation will be a very useful and eminently necessary if someone wants generalize the problem. Normally students are not common with the construction of graphical representations for functions but this ability can be caught up by simple examples. Several ways exit to succeed [8].
One way which we think is the best is the use of PROGRAPH diagrams [5]. The PROGRAPH-diagram for a generalized functional implementation of a circle and a diagram for the functions that calculate the intersection-points and a module that yields the output may look like as follows:

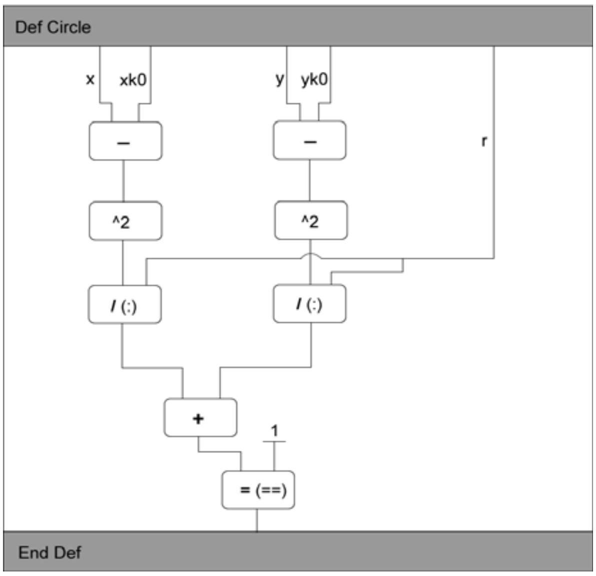


Fig. 3 PROGRAPH diagram for the function **circle**

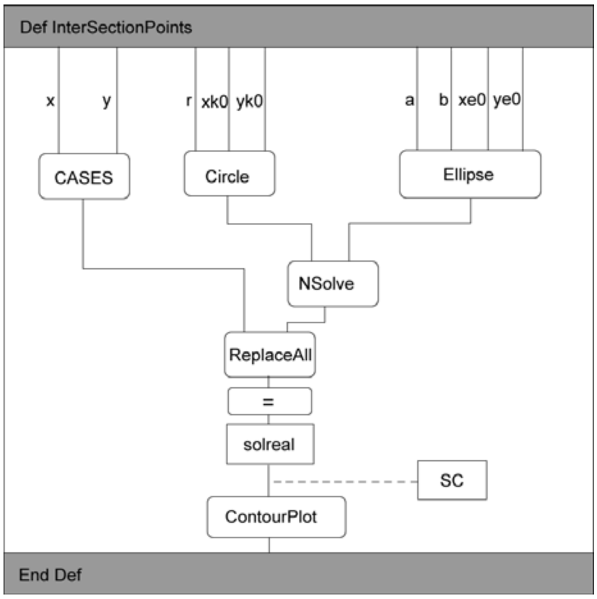


Fig. 4 PROGRAPH diagram for the function **Intersection-points**

In this diagram you can see an element (**SC**) which is linked through a dashed line with the rest of the dataflow - diagram. **SC** (note: SC means Structogram) the imperative-procedural module for the output procedure. We have provoked a most interesting situation: here two paradigms of informatical modelling are joined together. The Structogram of this module may look like the following:

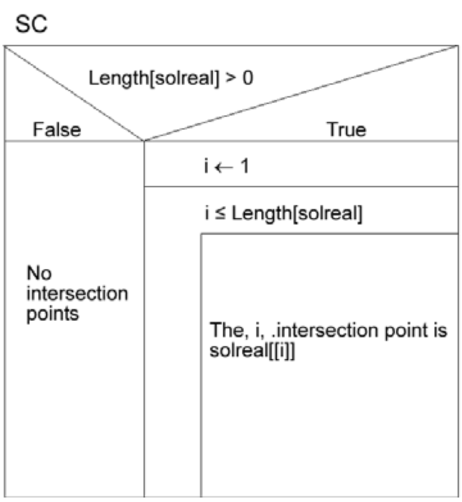
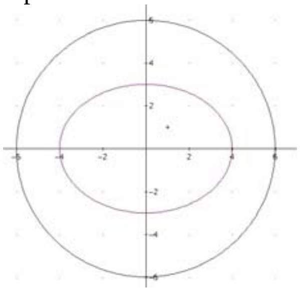
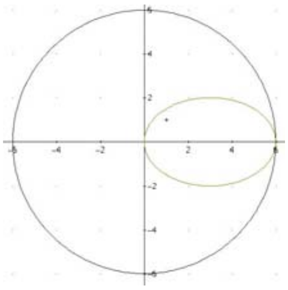
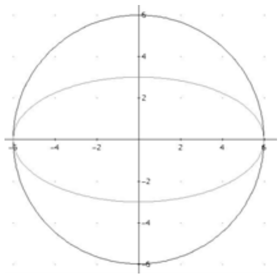
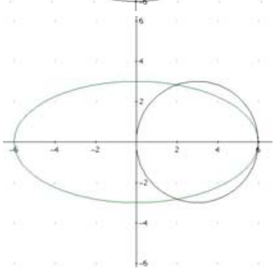
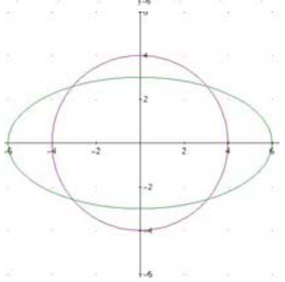


Fig. 5: Structogram **SC**

Step #3: (Functional) Implementation

After the graphical representation has been completed successfully the code can be implemented in the same way as we are using a tool which will be able to translate the functions.
First the students have to define the (implicit) functions for the circle and the ellipse:
Eq. (1): **Circle[x_,y_,r_,xk0_,yk0_]:=((x-xk0)^2/r^2+(y-yk0)^2/r^2==1)**
Eq. (2): **Ellipse[x_,y_,a_,b_,xe0_,ye0_]:=((x-xe0)^2/a^2+(y-ye0)^2/b^2==1)**
After the two functions are implemented the students can make experiments by variegating the parameters a, b; r, xk0_, yk0_ and xe0_, ye0_ and study their influence on the properties of the objects. The experiments lead to 5 possible states of the two objects:


a)

b)

c)

d)

e)
Fig. 6: 0 (a), 1(b), 2(c), 3(d), 4(e)
Intersection points.

As the functional code for the calculation of the intersection points (0, 1, 2, 3, 4) is very complex, customarily it will be prefabricated by the teacher. Both methodological ways are practicable, the way to ‘open’ the code and discuss it with the students as well as using the *function module* as a Black Box and studying and interpreting the different outcomes (Step #4) when variegating the parameters a, b; r, xk0_, yk0_ and xe0_, ye0_ again.

```
InterSectionPoints[r_,xk0_,yk0_,a_,b_,xe0_,ye0_]:=
    Module[{x,y,solreal},
    Kreis[x,y,r,xk0,yk0]=((x-xk0)^2/r^2+(y-yk0)^2/r^2==1);
    Ellipse[x,y,a,b,xe0,ye0]=((x-xe0)^2/a^2+(y-ye0)^2/b^2==1);

    solreal=Cases[{x,y}/.NSolve[{Kreis[x,y,r,xk0,yk0],
    Ellipse[x,y,a,b,xe0,ye0]},{x,y}],{_Real,_Real}];

    If[Length[solreal]>0,Do[Print["The ",i,".intersection point is:
    ",solreal[[i]]],{i,1,Length[solreal]}],Print["No intersection points"]];
    ContourPlot[{(x-xk0)^2/r^2+(y-yk0)^2/r^2==1,
    (x-xe0)^2/a^2+(y-ye0)^2/b^2==1},{x,-20,20},
    {y,-20,20},Epilog{PointSize[0.02],Hue[1],Point/@solreal}]]
```

In the (*functional*) module **InterSectionPoints[]** five parts are communicating among each other:

- 1st Part: Definition of the implicit function **Circle[]**,
- 2nd Part: Definition of the implicit function **Ellipse[]**,
- 3rd Part: Definition of the function for calculation the real intersection-points of the circle and the ellipse,
- 4th Part: Implementing an imperative-procedural module **SC** for creating a readable output,
- 5th Part: Definition of a graphical output function.

All of these five partitions could be started alone and would run without the help of the others. Through combining these defined functions they form an own enhanced function which can be discussed, modified or tested.

3. Conclusions

As we have seen Functional Programming combines the two subjects Informatics and Mathematics naturally and very intuitively. A very interesting point in this combination is the question of computability and calculability. This is a very old question which has long engaged scientists in Informatics as well as in Mathematics. The treatment of this question is a very theoretical. So it is very difficult to recommend it for education. But it will be possible to provide an insight into it if teachers are able to formulate adequate problems. We think that (functional) modelling is an adequate way to convey competences and to access more understanding for mathematical and informatical ideas.

References

[1] Dorninger, Ch. et al.: Angewandte Informatik – BHS Bildungsstandards – Das Kompetenzmodell, bmukk, 2008

[2] Fuchs, K. J.: Schulinformatik – Quo vadis? – Über die Notwendigkeit zentraler Leitideen. In: CDA 10/2003, pp. 18–19

[3] Fuchs, K. & Landerer, C.: Das mühsame Ringen um ein Kompetenzmodell. In: CDA 12/2005, pp. 6–9

[4] Hubwieser, P.: Didaktik der Informatik, Springer, Berlin, 2007

[5] Matwin, S. & Pietrzykowski, T. (1985): The Programming Language PROGRAPH: A Preliminary Report. in: Computer Languages, 10:2, pp. 91–125

[6] Peschek, W.; Heugl, H.: Standards für die mathematischen Fähigkeiten österreichischer Schülerinnen und Schüler am Ende der 8. Schulstufe, Version 4/07, Hrsg: Institut für Didaktik der Mathematik – Österreichisches Kompetenzzentrum für Mathematikdidaktik – Fakultät für interdisziplinäre Forschung und Fortbildung Alpen-Adria-Universität Klagenfurt, Klagenfurt, 2007

[7] Siller, H.-St.: Two Subjects Sharing One Concept – Functions As Structuring Principles in Mathematics and Informatics Education, accepted for ISSEP, 2008

[8] Siller, H.-St.: PROGRAPH Diagrams – A New Old System For Teaching Functional Modeling, accepted for ICME TSG 22, Monterrey, 2008

[9] Siller, H.-St.: Über die Bedeutung der grafischen Repräsentation beim Funktionalen Modellieren, In. Beiträge zum Mathematikunterrricht 2008, Budapest, 2008

[10] Siller, H.-St.: Informatics – A Subject Developing Out of Mathematics – A Review from 1977 to 2007, In: HPM-conference-proceedings, Mexico City, 2008

[11] Siller, H.-S.: Bildungsstandards - Mathematik – Version 2, Lehrerhandreichung, ph Salzburg, 2008