Free Software in education

Advise, vision and proposed action plan

Expert group:
Herman Bruyninckx
Mark De Quidt
Wilfried Feijens
Kim Lauwers
Eric Verhulst

Ministry of the Flemish Community
Education Department

Table of contents

Free Software in education
Vision and action plan

Herman Bruyninckx, Mark De Quidt, Kim Lauwers, Eric Verhulst, Wilfried Feijens

1. Introduction

This document develops a vision and an action plan for the responsible introduction of Free Software in education. It is based on existing (though not yet always recognised) needs and provides action points to support the different target groups in school practice. The essential contributions of this document are summarised in six recommendations for the responsible introduction of Free Software in education practice, each with short, medium and long-term suggestions.

This document not only highlights the technical potential of Free Software in education, but also its non-technical added value. The latter is evident particularly as a model for the development and cooperation for the meritocratic creation of, and democratic access to, knowledge and information.

For a good understanding of the terms and definitions used in this text, the reader is advised to consult the list of terms in the Appendix, or by following the hyperlinks (in the electronic HTML and PDF versions of this document).

2. What is Free Software?

This section provides a summary of the essential characteristics of Free Software, better known internationally under the names *Free Software* and *Open Source* software. For all the details we refer to the website of the Free Software Foundation and of the Open Source initiative.

2.1. Definition

Free Software is software in which the makers (i) make the source code available and (ii) use the protection of their copyright to link a licence to this source code which permits free use. The software can be used free of charge and without explicit consent on an unlimited number of computers and can be adapted, disseminated and integrated with other software. This *free use* goes much further than the fair use for which regulations have been introduced in the legislation, and in practice avoids the lock-in effects which are so damaging in closed software.

However, the *free use* is not unlimited, because the users must observe a number of conditions when adapting, disseminating or integrating the material; these conditions are described in the licence which accompanies the software. Every Free Software licence imposes slightly different conditions in this field and in practice, these conditions are somewhere between the extremes represented by the following licences:

1. GPL: adaptations which are disseminated fall under an obligation to return the software and integration is permitted only with software for which the licence is “GPL compatible”.

One example of GPL software is the Linux-kernel (Linux is a Free Software operating system which has been very successful worldwide in all the ICT sectors since the late 1990s).

2. BSD: adaptations do not have to be returned, and the integration with software with any other licence is permitted.

An example of this sort of software is the Free BSD operating system, which Apple uses in its Mac OSX products.

In the context of ICT in education, it is important to realise that the above- mentioned concepts of *available source code and free use* not only apply for software, but for all immaterial forms of knowledge and information, such as texts, sound, images, figures, etc. Specific licences also exist for these, such as, for example, the Creative Commons licence.

2.2. How is Free Software created?

The democratisation of the internet creates unlimited possibilities to immediate worldwide cooperation in a community of interested parties. This technological opportunity has taken advantage of the so-called “creative mass” to prove that it can create socially relevant added value (in a cultural, but also in an economic sense!), outside the traditional economic actors of the *market* and the *company*. This occurred not so much because it was technologically possible, but above all because many people and businesses benefit from it. Furthermore, the Free Software licences (with GPL in the lead) provide the legal framework to achieve the best possible balance between, on the one hand, the personal enrichment of every individual in that creative mass, and on the other hand, the leverage effect which occurs in a community when everyone makes the knowledge they have developed freely available to that community.

Free Software is such a “lever”, particularly because of the high level of modularity: it is self evident that new projects make as much use as possible of already existing components (to save effort), and in turn produce modular software in the hope of being used by as many other projects as possible (to flatter the developers). In other words, the wheel is not reinvented every time. Or at least this applies in principle because practice has shown that there are enormous numbers of Free Software projects which just produce yet another new version of already existing ICT applications. Most of these *Me too* projects do not provide direct added value, but are indirectly still an essential component of a healthy Free Software biotope, because (i) they give newcomers the chance of gaining experience and building up a reputation, and (ii), they create a “biodiversity” which cannot be paralleled by commercial software producers.

2.3. Why do people make Free Software?

The term used above "personal enrichment" refers both to individual people and to companies, and means much more than purely financial benefits: many volunteers have a sense of satisfaction when their contributions are valued by others and when their name acquires a reputation within a community of peers. Furthermore, for many businesses, Free Software is a business model based on sound arguments which is actually not very different from the model behind many traditional professions: after all, a plumber, surgeon or lawyer also performs in such a way that the results can be examined, used and adapted by anyone (or usually by peers), without affecting the intellectual property rights of the original professionals.

2.4. Free Software and education

The distributed and incremental development model of Free Software not only works with software, but for all abstract forms of knowledge and information, insofar as this knowledge and information benefit from (i) incremental adaptations, and (ii) discussion in a large community of peers. Therefore the Free Software role model in principle also works for most types of educational materials: text books, educational *content*, catalogues of *best practices* in school practice, training for teachers or ICT coordinators, etc. Which commercial provider of educational materials and software, for example, in mathematics or French could compete with a motivated group of several dozen teachers cooperating in these fields? Particularly when you take into account the fact that international cooperation in Free Software is almost self-evident. In fact, in practice this cooperation is much easier to achieve than that between publishers of educational materials who all want to protect their own intellectual property rights.

The lever of Free Software will probably not be efficient in fields such as art and culture. In these fields there is a possibility of reusing ideas, but not of improving the works of art themselves incrementally or with group discussions.

2.5. Competitive advantages

1. Free Software can be used free of charge. However, nothing prevents the makers of Free Software from asking money for their creation, though the freedom with regard to unlimited copying automatically leads to an unlimited downward price spiral when there is a large group of users.

   N.B.: The *use* of the software may be free of charge, but the *services provided* in relation to Free Software (creation, installation, maintenance, integration, programming, training…) does not have to be free at all. Therefore Free Software allows for the possibility of developing commercial activities based on the software.

2. Free Software generates human networks of interacting users and makers, because everyone has access to all the information about the project (documentation, code,

design discussions, problems, etc.). It is these networks which give Free Software its large leverage. For example, this lever is quite frequently more efficient than commercial *helpdesks* to solve problems, and also goes much further towards a constructive participation of the user in the evolution of the software.

It should be remembered that the large majority of Free Software projects never achieves the critical threshold of interested users to achieve this leverage affect.

3. Free Software never goes bankrupt and therefore does not “destroy” any knowledge. After all, in contrast with commercial software, the code and knowledge in a Free Software project do not disappear when its makers cease their activities for whatever reason.

   In practice, it often happens that Free Software projects are not further developed (in fact, this is even the case with a large majority of projects), but good ideas and the code from these projects can be reused in other projects without any problems.

4. In practice, Free Software leads to the integration of different projects and initiative, while commercial software often leads to “extegration” in monolithic programmes and counterproductive differentiation.

   Integration is very important in ICT (and also in all the sectors outside this), because it permits different programmes to work together, independently of the supplier or computer platform. However, commercial ICT practice does not naturally tend towards integration. All the manufactures make as many efforts as possible to integrate only their own programmes and to make the integration with products of competitors as difficult as possible (this is known as “extegration”. This unsatisfactory exchangeability greatly increases the chances of monopolies in the whole ICT field.

   The commercial UNIX market is an example of counterproductive differentiation: every provider has developed his own version of UNIX to differentiate himself from the competition. The result of this is that none of the commercial UNIX versions have access to all the innovations that have been developed, the software is not platform independent, and the price is therefore unnaturally high. This has resulted in the gradual disappearance of commercial UNIX versions from the ICT market, to the disadvantage of Windows and Linux.

   Free Software projects particularly benefit from cooperation, which means that the leverage effect of Free Software can be achieved as far as possible. Integration is also much simpler because Free Software gives preference to the use of open file formats and communication protocols.

5. Free Software never places users before a fait accompli, although this is the norm for commercial software: after all, most businesses do not wish to invest more in

versions of the software which have been sold as soon as they have newer versions available.

Free Software is user-friendly in this respect because (i) it enables everyone to maintain existing software themselves, even if no one else wants to do so, and (ii), it is possible to *upgrade* to new versions without licensing costs and without licensing administration.

6. Free Software has a strong UNIX background and therefore uses strong modular software components.

   This modular approach leads to systems which can be maintained more efficiently, to greater flexibility in making specific configurations, a better controlled security and less far-reaching consequences when *security leaks* do occur after all.

7. Free Software is based on an ethical code and a mentality which perfectly complement the values and practice of education: the unconditional sharing and passing of knowledge, critical reflection, cooperation, a respect for other people's creations and views, etc.

   In practice, this leads to highly motivated employees, because the sense of being socially useful and ethically responsible are strongly present in most Free Software projects. This is the same motivation which has encouraged people for centuries to participate voluntarily and without payment in charitable activities, sports, clubs, youth movements, music workshops, etc.

   Free Software is accompanied perfectly by commercial software: more and more ICT suppliers are supporting their products under Linux, without having to release them under a Free Software licence.

   The most common "type of cooperation" is that Free Software is responsible for the "lower layers" of an ICT system, while closed software deals with the "higher layers" where there are more pickings in terms of added value. The evolution is that Free Software has moved higher and higher up this chain.

2.6. Competitive disadvantages

The above-mentioned advantages have a permanent character while the disadvantages discussed in this section are only temporary. However, for many users, these disadvantages are still substantial at the moment and it is very difficult to predict how long the period will last in which these disadvantages will continue to apply.

1. At the moment there is still little educational material available in Free Software, and its quality is also lower than the commercial provision, and even lower than the freeware provision.

2. Most people see Free Software as a brand new, incompletely understood (r)evolution, and consequently have a rather conservative and critical attitude. There is a great deal of inertia against change in their ICT habits.

3. Before a Free Software project can be established in a particular field and can develop and remain viable, initial starting efforts must be made in some way. For "small" target groups, commercial initiatives are better and faster, at least under the condition that the target public has adequate financial means itself.

4. At the moment, the ICT market still provides much less commercial support for Free Software than for software on Windows and MacOS, except for the large projects (Linux, Apache, Mozilla, Openoffice.org, etc.).

5. Every newcomer to Free Software is confronted very quickly with apparent chaos: the range is so overwhelming that is easy to lose your way when you first come across it.

   The Linux distributions (Red Hat, Suse, Debian…) are already providing ready-made solutions without the above-mentioned chaos. The Dutch foundation, ICT at school has made a CD package available for education.

3. Target groups

"Education" is not a unique, uniform target group with identical needs. A description of the relevant target groups and their specific needs and greatest challenges is given below.

3.1. Policy

(Ministry of Education, organising authorities and umbrella organisations).

Needs: the best possible ICT for as little money as possible, good provision of education for other educational target groups; guarantees for an honest and transparent ICT market and for a minimum of "biodiversity" in the ICT solutions used; establishing international networks; the identification of the relevant ICT skills and recording them in attainment targets; providing sufficient hardware for schools.

Challenges: defining the "correct ICT skills" in attainment targets; as identified by the educational ICT coordinators; finding a balance between monitoring and stimulating ICT developments with which teachers can achieve these attainment targets; ensuring the

correct place for Free Software in education; finding a balance between saving costs and dependence in *Public-Private Partnership* related to ICT.

3.2. School administrations

Needs: finding the best possible ICT support for the specific education-related ICT processes; achieving a critical mass and synergy in this field with other government administrations to ensure the right place for Free Software and, in the first place, open standards. After all, the ICT needs for Education are not fundamentally different from those of other government administrations.

Challenges: to ensure that: (i) all administrative ICT works with open standards as quickly and extensively as possible in order to create a fair and competitive market, and (ii) the administrative ICT solutions do not inhibit the efficient integration of Free Software in the other target groups.

3.3. Technical ICT coordinators

These are the “people in the field” who, together with the educational ICT coordinators and teachers, play a key role in the successful introduction of ICT in the classroom.

Needs: the capacity to install and maintain ICT networks; monitoring the evolutions of the (free) software market.

Challenges: learning to discover the competitive advantages of Free Software and implementing this in school practice without being constantly distracted by the very short-term solutions which the majority of administrative and educational staff expect. The technical ICT coordinators must prepare to play an important, probably even the most important role in every change in ICT practice in education.

3.4. Educational ICT coordinators

These are the “people in the field” who must support the teachers, with the educational aspects of the use of ICT in teaching practice.

Needs: gaining experience in connection with how and at what age ICT can best be integrated in non-computer-related subjects; identifying the instrumental, useful and self-development ICT skills which children can tackle at different ages; looking up and/or providing additional support for software and teaching materials; participating in (international) networks.

Challenges: identifying ICT skills; creating appropriate teaching materials.

3.5. Teachers

Needs: *ready-made* teaching materials with good support, which can still be adapted and incorporated in the existing curriculum; training, both with *supervised self study* and in traditional form. Using Free Software where possible in order to provide software aids to all pupils cheaply and on an equal basis.

Challenges: learning to cooperate effectively to produce “educational content”; understanding and learning to use Free Software alternatives.

3.6. Teacher training

Needs: training, development and explanations for concrete teaching materials, so that they can be used responsibly and constructively with ICT in as many subjects as possible. Identifying and discussing ICT attainment targets (exactly what can be done and at what age, what is the real added value of ICT, etc.); cooperation with institutes of higher education and universities.

Challenge: overcoming resistance to change; achieving a critical mass to achieve training materials with Free Software.

3.7. Pupils

Needs: teaching the essential ICT skills; discovering Free Software (in the broad sense of the term), as a role model for independent and lifelong learning.

3.8. Commercial providers of educational materials

Needs: the availability of a powerful common *pre-competitive infrastructure,* in which they can implement their own added value without the obligation of releasing this added value as Free Software, and also without each having to make the large investments of producing and maintaining that infrastructure. The basic requirements of the infrastructure are: a high degree of modularity; support for the configuration of modules by the teacher; the exchangeability of modules and data between different suppliers; support for several languages; the configurability of figures, keys, photographs, etc.; (web) *client server* architecture which allows for the use of light terminals, etc.

Challenge: finding the suitable software design and architecture for this infrastructure and collecting a sufficient international critical mass for this purpose.

4. Vision

This chapter describes the direction in which ICT should develop in education, for what reasons and the place of Free Software in this evolution. The next chapter gives suggestions about how this vision could be put into practice. This takes into account the above-mentioned needs of all the target groups concerned.

The proposed vision is not exclusively related to Free Software: most ICT skills can also be taught without Free Software. The competitive advantages, together with the fact that Free Software devotes much more attention and support to “object-oriented ICT skills” because of its UNIX past, mean that achieving an educational project related to the intended ICT skills is much easier. Obviously the chances of cheaper solutions are also much greater with Free Software.

4.1. ICT skills

Target groups: policy, teachers, teacher training, ICT coordinators.

Why? In the final analysis, ICT technology provides few or no really new possibilities in education, industry or administration. In principle, ICT does allow for an enormous enlargement of scale: old working methods can be greatly speeded up, many more people and information can be accessed in a much faster and cheaper way. But ICT is still a new technology. In fact, we do not yet know exactly what the essential ICT skills are which should be taught at every level of education, or at what age or in what way these skills can best be taught. Therefore it is important to adopt a modest view in identifying these ICT skills. In the first instance there is a need for a thorough analysis of which traditional working methods are eligible for ICT support and exactly how these traditional working methods can be improved by ICT. All the levels in education should be aware of this situation, accept that a great deal more exploration must be done (with guaranteed frustrations as an unpleasant side effect) before the list of ICT skills, including their educationally most justified support, is sufficiently understood to be translated into attainment targets.

Vision. The top ten possible directions for exploration are given below, where possible with an indication of where Free Software can be used as a successful field of experience and/or as a source of software for concrete support in school practice. (More concrete software examples and supporting actions are shown in the Action plan).

1. Collect concrete teaching materials to explore ICT skills. The administration of the Ministry of Education has already taken a valuable first step in the desired direction, as revealed in the recent publications ICT competences in primary education and ICT on the menu. The policy should continue to actively support these efforts, amongst other things by organising workshops related to these types of concrete exploratory activities. It is only on the basis of a permanent and critical discussion with all the target groups involved that these initiatives can be sufficiently representative and achieve results sufficiently quickly.

   Free Software ICT practice can also serve as an example in taking the following steps: a large number of the ICT competences (or skills) that are aimed for are taught “naturally” to newcomers by the Free Software community. This concerns skills such as the critical use of ICT resources and looking for possible alternatives to familiar programmes. It is precisely the last aspects of being able to

be learn and evaluate alternatives that is vitally important in the search for the definitive list of essential ICT skills.

2. A strict division between storage, processing and design. Almost by definition, ICT is extremely suitable for the automatic processing of information. However, the great majority of current ICT users are stuck in the so-called “user friendly” WYSIWYG paradigm and in a monolithic software lock-in, which confuse the distinctions between the three most important components of efficient ICT use: (i) the format for electronically storing data, (ii) the software for processing the data, and (iii) the form in which the data are displayed to the user.

   If the format of stored data (for example, illustrations or text) is understood, it is also possible to use programmes of other suppliers and a different design. The interconnection of storage, processing and design. as well as the secrecy regarding these processes form the core of some of the dominant commercial suppliers. This results in a financial, technical, cultural and aesthetic impoverishment of the ICT landscape.

   Free Software is not affected by the above-mentioned commercial pressure and can therefore experiment with alternatives without any restrictions. In fact, this “sense of exploration” is one of the reasons why university ICT research is so often available under Free Software licences, even long before the breakthrough of the PC and the internet.

3. It is not at all necessary to abandon the WYSIWYG paradigm, but education should indicate alternative ways of dealing with electronic information and clearly show in which situations the various methods are most suitable.

   Free Software provides an enormous range of these alternatives, inter alia in the form of scripting languages, client-server programmes and commando line programmes. Concrete examples are given in a number of the paragraphs below.

4. Creating information = the structuring and semantic identification of data. The progress of electronics means that it is possible to produce and exchange and enormous amount of data. However, data without identification and structure are not very valuable as *information*. Therefore it is important to familiarise pupils with the basic characteristics of structured data (databanks) and with the motivation why these should be used much more instead of, for example, spreadsheets. After all, the latter only work on the PC of individuals, work with flat “spaghetti” data structures and make it difficult to share information which transcends individuals and maintain its consistency.

   In addition to structuring information, semantic identification is the second important component in dealing with information: information is only useful if its meaning is clear and the same information can have a different meaning in a different context.

The aim to familiarise pupils with the structuring and semantic indication of information can be achieved without explicitly being concerned with databanks or ICT. After all, the essential skills of imposing a structure and assigning a meaning are the same as for more everyday problems such as writing a report, organising and managing a large group project, finding structure in large amounts of information in biology (the names of muscles and bones; the classification of animals and plants; etc.) and so on.

5. To promote text editors as the real basic tools for all ICT needs, both when entering data (for example, the text of a report) and for the structuring of those data into information (for example, the "semantic tags" in the HTML language which encodes the information and the form of web pages). Free Software provides a whole range of text editors with different levels of possibilities.

   Teaching a text editor requires some effort, but it is an instrument which has proved to be useful day after day and on a permanent basis. This applies particularly to text editors who do not need a mouse, because the user gains a great deal of time by avoiding the movements of the hand between the keyboard and the mouse, and vice versa.

6. Insight into the basic technology of the ICT: processor, memory, periphery, network (computer addresses and names; servers, security problems; etc.); the relevance of open standards, etc. A (passive) understanding of technology is a necessary condition for removing the fear of ICT evolutions in users and to give them sufficient self-confidence to be able to act as a critical ICT consumer. This is exactly the same for all other technical evolutions in our society, with the difference that the average pupil or teacher is familiar with the backgrounds of, for example, the car, electrical appliances, the telephone, the radio, the newspaper, etc.

   An introduction to the technology does not have to be detailed or take place in specially organised subjects. However, it is essential to increase ICT self-confidence.

7. An insight into the basic file formats used for the storage and exchange of information with the computer. It does not take a great deal of time to explain when information can best be stored in a word processing format, in a databank, in a PDF file, in an HTML web page, etc.

8. Familiarity with the basis of "programming". This intimidates many pupils and teachers, but basically programming is not much more than formulating ideas in a structured way and can be applied to so much more than just to software. You can teach pupils "programming" skills by getting them to organise a large (non-software) project, asking them to explain how laws are set up and implemented in Belgian politics; by asking them to draw up timetables; etc.

9. Learning to use the web services. The future classroom and future workplace will above all make use of information storage and processing on centrally managed servers with a user interface via a web browser or a portable “mobile telephone”. After all these technologies are strong enough to support the majority of ICT needs. Furthermore, centrally managed ICT applications require much less time from the ICT coordinators; the installation and maintenance of one server, and not much more than one browser on everyone’s computer.

   The most important contribution to education of the recent evolutions of the internet is undoubtedly the following: the rapid and unrestricted access to information has become more important than the traditional *possession of information*. Instead of gaining all the information from information sources which are available in the school library, modern pupils and teachers benefit from knowing how they can consult the correct information sources on line and where they can communicate with *people* who can provide an interpretation and structure for these information sources. Obviously these on line possibilities must be used together with the traditional libraries.

   The essential ICT skill in this field of “web services” is certainly not the ability to technically create the web infrastructure, but learning to see which components of automatic information processing can best take place on the *server*, and which things are better dealt with by the *client* on a personal PC or *handheld* computer. After all, the more work and functionalities that can be moved to the server, the more efficient the maintenance of the ICT infrastructure, the cheaper and lighter the clients, and the more universally accessible the information.

   Most commercial software producers try to inhibit the evolution to the web services as far as possible because (i) Free Software is enormously strong in these fields and therefore results in too much competition, and (ii) it is much more difficult for commercial software producers to sell individual licences if their applications (ii) it is much more difficult for commercial software makers to sell individual licences if their applications run as web services without clearly identifiable “clients”.

10. Ethical and social aspects of ICT. Information is not only economically useful, it also has an important ethical and social component:

    - some forms of information must be well protected because of copyright and rights to *privacy*
    - because of the lock-in phenomenon, the commercial ICT market is increasingly dominated by a few mammoth companies which can deny less capital- intensive, but very often interesting and creative companies a fair place in the market with the lock-in process.

- The concept of intellectual ownership is in danger of being reduced to a legal dual with regard to the ownership of patents and Digital Rights Management.

Therefore pupils must learn to deal with the sometimes vague distinction between fair use and abuse.

Moral and ethical aspects apply with regard to the way in which the "owner of information" who makes information available to the "person requesting information": selling it, leasing it with a user's licence, providing it freely under a Free Software licence etc. With regard to education, this information primarily concerns knowledge, teaching materials, documentation for training etc.

A large proportion of the Free Software community has a very strong ethical and social awareness, as reflected in the first place by the motivations of the Free Software Foundation. With regard to education, the FSF and other Free Software initiatives are very clear and concrete role models which are certainly worth discovering and examining.

At first sight the above-mentioned skills appear to be at a rather advanced level, but they complement the ICT competence diamond drawn up by the administration of the Ministry of Education perfectly. This diamond comprises the following nine aspects:

1. Cooperating on a task
2. Presenting information
3. Independent learning with the help of ICT
4. Searching for and processing information
5. Communicating information
6. Practicing with the help of ICT
7. Being creative with the help of ICT
8. Instrumental skills: using the equipment
9. Social-ethical competences: the pupils use ICT appropriately and in a responsible way.

4.2. Free Software as a model

Target groups: teachers, ICT coordinators, teacher training

Why? Even the teachers who are most highly specialised in ICT and ICT coordinators in education are not very familiar, or totally unfamiliar with the leverage effect which the Free Software model can have in the creation, improvement and dissemination of educational materials, and how the Free Software community achieves the joint design and development of projects in practice.

Vision. The policy must ensure that these unique advantages of scale and the resulting win-win situation are made understandable for all teachers. Essential aspects are a good understanding of:

- the concept of a Free Software licence. This is not in the first place in a strictly legal field, but in the field of understanding why these licences create a win-win situation to stimulate both making available intellectual efforts and extending and improving the result. After all, Free Software licences guarantee that the authors are always adequately recognised for their efforts, while they also guarantee that developments and improvements find their way back to the community.

- how Free Software projects make use of the internet to support their activities as far as possible: website, discussion groups, databank with material made available by *web services*. Therefore Free Software is not only the most convincing practical example in itself, but also has all the technical infrastructure and voluntary and/or commercial support.

The Free Software model is simple, does not require any technical baggage to be understood and can be presented with examples which are not related to ICT in any way; for example, the creation of course texts or exercises for mathematics or language. It is even *appropriate* for the policy not to particularly provide incentives for software projects, but also for all the initiatives which are related to the creation of educational materials in the broad sense of the term: courses, multimedia, exercises, etc.

For education, Free Software has a unique competitive advantage in stimulating and supporting basic initiatives. This means that individual teachers who have ideas for educational projects can achieve a great deal themselves on the basis of the codes and documentation available for other projects. At the moment, most projects by teachers do not make use of Free Software but do often use freeware. However, freeware has the disadvantage that source code is not available, and this restricts the incremental expansion and improvement of the initial initiatives so that the advantages of scale are completely lost.

4.3. Stimulating international networks

Target groups: teachers, ICT coordinators, teacher training, policy

Why? The ICT needs in education are extremely universal in the sense that they do not differ drastically from country to country, or from one language area to another. The differences are at the level of language, the level and selection of programmes; timing during the school career; intensity of use; etc. This diversity can easily be absorbed with the configuration of software programmes. This configuration in response to regional or national needs can only be achieved on the basis of a strong modular character of both the software infrastructure and of the (international) development community. Intelligently designed software programmes (or other abstract educational materials) can greatly benefit from international cooperation because the group of contributors with the

necessary expertise and motivation increases with the number of countries for which that programme could be interesting.

Vision. International cooperation at the level of developing educational Free Software programmes and documentation is essential. All the parties involved in education should be stimulated to participate in this international cooperation or to establish it where necessary.

In all the new educational Free Software initiatives, priority should be given from the start to a modular approach, and language, platform and cultural neutrality. If possible, adaptations to local needs should be made on the basis of a simple configuration.

With regard to modularisation and language localisation of software and documentation, Free Software has more than an edge, as can easily be seen in the approach of the commercial Linux distributions. They only have a small number of personnel and are responsible for only a very limited share of the software and documentation which ends up on the CDs, Furthermore, the majority of the programmes are available on the same Linux CD in dozens of different languages.

The services with which Linux distributors manage to survive commercially, consist of the selection and integration of thousands of software modules produced by the Free Software community, without any form of strictly hierarchical development. Therefore there are sufficient expertise and "role models" for educational projects in Free Software. Free Software developers are completely convinced of the value of independent, modular components which can be integrated because they work with, and have respect for standardised interfaces.

4.4. ICT infrastructure

Target groups: ICT coordinators, policy, school administrations

Why? The technical aspects of the ICT infrastructure are similar in all schools, and therefore it is obvious to share expertise and solutions. However, the current situation is still a long way away from this objective. This is by no means least because schools still often think in terms of a "personal computer"; i.e. a computer where all the software is available in situ. This makes the installation and maintenance of all the computers very time consuming in comparison to working with networked "thin clients".

Vision. Where possible, ICT in schools should evolve towards centrally managed applications which can be used across local networks on the *thin clients* of the pupils, teachers and administration. This architecture requires much less manpower for maintenance. Because of its past, Free Software has always been extremely suitable to run centrally managed programmes with *thin clients*. Free Software supports older hardware much better than any other commercial alternative and all the infrastructure and management programmes are available for a networked ICT infrastructure. As there are no *licensing costs* for Free Software, the uniform and networked installation and the

upgrading of all the computers can be carried out very simply and cheaply. In addition, Free Software is inherently much less sensitive to *viruses, spyware, adware*, and other software that is not well-intentioned.

The previous paragraphs deal with the *hardware and software* infrastructure, but every ICT system also requires a suitable human environment: a network of experts who exchange information and best practices. They can even partly and/or temporarily take over each other's tasks, either in situ or from a distance.

The policy should stimulate the development of these human networks with the technical ICT coordinators and aim for the organisation of first and second lines of support: the more specific and detailed certain ICT skills become, the fewer people will have these skills, but the more efficiently they will be able to use these skills on larger ICT networks. These most specialised ICT coordinators must be able to work from the second line in a "screened off" way, i.e., as support for the first line ICT coordinators who in turn help the administrations and teachers. The large Free Software projects can also provide many role models for these human networks. For example, the developments of the Linux kernel, OpenOffice.org and the Mozilla browser take place in a similar hierarchical way.

4.5. Promoting a vision

Target groups: ICT coordinators, policy

Why? All the ICT-related actions of the Ministry of Education should be part of a long-term vision of ICT. The current vision document is too vague and too general and does not have the courage of its visions for companies and society. The first reflex in the education policy is still all too often "*We are listening to the direct demands of companies and society*!". However, in this way, education will never change anything in the inadequate ICT skills and the fairly uncritical attitude to the prevailing ICT market. Free Software has a large number of competitive advantages in this field and education should be the first to recognise and promote these.

Vision. A vision must be a real vision, which means that there should be a number of long-term objectives based on sound reasons which are reflected in the actual policy as fully and completely as possible. In concrete terms, this means that at the moment, more than just the vision must be defended in this document against a number of deep-rooted ICT ideas, both in industry and in society, and in education itself.

Therefore education has *nothing* to lose from a positive discrimination by Free Software:

- every (non)-investment in education has macro-economic and social consequences for a whole generation.
- the so-called *Total Cost of Ownership* of ICT investments in education has a very distant horizon.

- because the policy on ICT at school must be guaranteed in the long term, it is clear that the introduction and stimulation of Free Software should start as quickly and intensively as possible.

This action should be given priority over the current initiatives.

In fact this “positive discrimination” is not discrimination at all with regard to commercial ICT providers: the evolution towards the general use of Free Software for all low level ICT infrastructure is irreversible and can be perfectly justified economically. Commercial providers can concentrate on the creation of real added value on a neutral ICT platform which does not benefit or disadvantage anyone.

At the non-economic and technical level, it is also justified to promote the Free Software model in education as a priority: no other alternative can put an ever-increasing number of people directly into contact with the constructive process of creation and improvement of Free Software. It is a certain and cheap way of achieving a better knowledge economy. The possible “disadvantage” of the open character of the Free Software model is that our “competitors” will also be able to benefit worldwide from the same unrestricted access to the knowledge economy.

5. Action plan

This chapter discusses concrete actions points which help to achieve the vision described above.

5.1. General actions

The execution of this action plan, in full or in part, must take into account the human tendency to oppose change. This is very strong in the education world, where teachers have very different concerns from dealing with “yet another change”. Especially when this is in a field (ICT) in which only a small minority of teachers really feel at home.

Therefore the only approach which can be successful in the long term is one in which there is no obligation, but with positive stimulation in the following ways:

- provide independent and realistic information. The government should not release money to create or structure this information itself (there are enough professors and assistants to do this in our institutes of higher education and universities), they should simply make the existing information accessible.
- ensure that in the administration of the Ministry of Education, there are one or two people who closely follow the evolutions in the Free Software world and make sure that they have intensive support of advisory groups from institutes of higher education, universities and industry. These people in the administration also coordinate the active selection of relevant information and its dissemination to the people in schools. These people are already present in the administration of

the Ministry of Education; they just have to be supported in a more explicit and visible way to make Free Software a priority.

5.2. ICT skills

The educational support of teaching materials for ICT skills in the chapter on Vision is undoubtedly the most important objective of all ICT efforts in education. Therefore the policy should give priority to actions to create a great deal of this material in the short term.

The first step is to motivate small groups of teachers and ICT coordinators with creative ideas for these sorts of teaching materials. The policy should put these working groups into contact with interested parties from institutes of higher education and universities. This cooperation must be proactive, with a targeted campaign in the education media. (It should not be forgotten that it is important to support the positive responses to this campaign with suitable ICT infrastructure.)

One obvious initiative which is fairly easy to achieve is the creation of specific teaching materials for "supervised self study" courses on the use of *HTML and Cascading Style Sheets (CSS)* for web and open standards-based word processing. Because of its universal accessibility via the web browser, this technology is also eminently suitable for providing most of the other teaching materials that are created.

There are already a modest number of examples of Dutch teaching materials available under a Free Software licence, for example, hier, hier en hier. However, a great deal more will have to be done before the "top ten" of the aimed-for ICT skills, are supported with educational materials.

5.3. Free Software as a model

As indicated above, it is the responsibility of policy to introduce the methods of Free Software projects as quickly and widely as possible in education, starting with the educational support for ICT skills described in the previous Section. A number of examples of relevant actions are described below:

- to stimulate and coordinate the initiatives which develop from the bottom up along the lines of the large Free Software projects. This means that every initiative must have a "natural" and respected leader, who reflects the vision of the project and directs its development. All this is always in consultation with all the interested parties.

- one of the necessary components of Free Software projects is the electronic forums for the exchange of information and experiences between project leader(s)

and interested teachers and “outsiders”, such as parents, academics, companies, etc. (These forums are part of the ICT infrastructure to be developed.)

- to facilitate the recycling of software infrastructure across different projects. After all, the available resources are scarce, and it should not be necessary to reinvent the wheel every time. This recycling does require expertise in the modular design and integration of software or documentation systems. This is a suitable responsibility for the participants in universities and institutes of higher education.

- the emphasis should be on web-based developments, both because of the technological possibilities and the general accessibility to all computer platforms.

- all the development of the didactic materials and software paid for by education must be given a Free Software licence. One obvious choice is: GPL for end-user programmes and LGPL for supporting libraries.

- all these actions must be sufficiently well known in education and society. This can be achieved, for example, with regular reports about Free Software and Free Software projects published on the Education website or through the existing publication channels. It can also be achieved with reports by all the projects that are started on their objectives, problems and achievements in six monthly workshops, where these projects can come together. A link with existing ICT trade fairs which is already being supported by the government (the Flemish education days, and so on) is an obvious step.

- furthermore, these reports should also be sent to the further training courses of REN Flanders on a permanent basis. Therefore it seems appropriate to permit schools to use their training budgets to allow teachers and ICT coordinators to participate in Free Software projects and workshops.

The contribution of teachers and those responsible for teacher training is essential, because most of the existing Free Software projects often miss their educational goal because they did not develop on the basis of specific educational needs, but as a result of the educationally less supported interests of technical developers.

In many of the above project suggestions there are also possibilities for *Public-Private Partnership*: all ICT suppliers (outside the scope of Microsoft: IBM, Novell, Sun, HP, etc.) just want Education to use and promote ICT in a way that is independent of supply.

5.4. Stimulating international networks

Whether or not it is for Education, Free Software has an international dimension. This networking is easiest with the already existing European contacts of the Education Administration. These people merely need slightly more visible and explicit support to

enable them to make this a priority. Obviously cooperation with the Netherlands is the most obvious aspect.

From the point of view of *Public-Private Partnership*, it is appropriate to make contacts with the commercial Linux distributors with strong European roots, particularly SUSE and Mandrake, via the above-mentioned European networks. These are obviously very interested in providing their products to as many European education organisations as possible and will certainly want to invest in educational Free Software projects.

5.5. ICT infrastructure

The development of an ICT infrastructure in relation to Free Software comprises a number of complementary components:

1. For ICT use in school practice, the priority is to develop server networks and thin client infrastructure for each cluster of schools, on the basis of Free Software components. (A "cluster" is the neutral term for every group of schools which are sufficiently close to each other to be able to share the network). This action was motivated by the savings expected for maintenance and licence costs, and by the better support for the aimed-for ICT skills.

   A second action priority is to integrate the above-mentioned educational materials in the large "*Linux desktops*": Gnome and KDE both already have a number of educational projects in their portfolio but there are still too few teachers working with them, and the integration of the educational material is still below par.

2. The most important infrastructure for the support of educational projects consists of making available centrally managed electronic forums. In practice this does not have to mean much more than, for example, a number of servers, with a mailing list, a databank with all the project data and a web page with information about the project for each participating educational Free Software project. The whole of this infrastructure can be created perfectly with Free Software.

   There are already a number of these electronic forums which were established on the personal initiative of a few teachers. However, almost all these forums are spread across free providers and are therefore very hidden on the Internet, polluted with advertisements and/or locked away behind registration procedures with commercial ends.

5.6. Promoting a vision

A permanent, positive and dynamic promotion of a vision, no matter how justified, must be carried out by well-informed and assertive employees. The policy must try to discover these employees in the community of people who worked on all the previous action

points. It is self evident that the academic partners have an important task promoting this action.

6. Recommendations

The recommendations in this chapter are a concise summary of the arguments and clarifications which were discussed in the previous chapters. Therefore the reader should not interpret these recommendations independently, outside the context of all the previous chapters.

The recommendations are not structured according to the different target groups in education, but in accordance with a number of complementary aspects of ICT school practice: personal self-development, Free Software as a model, ICT skills and "attainment targets", infrastructure, teaching materials and training. There is no order of importance based on any criterion in this structure or in the order of the recommendations: all the recommendations together form a coherent body, and each focuses on a different aspect of ICT in education.

The recommendations (except for Recommendation 0) comprise three levels, which require support by policy in the short, medium and long term:

1. To stimulate. A stimulating recommendation assumes that a great deal can be achieved by providing motivated people "in the field" with adapted and structured information, or by giving them clear signals that certain efforts and evolutions (in this case, the introduction of more Free Software in education) is by definition desirable and appreciated.

2. To facilitate. A facilitating recommendation goes a step further and requires that the stimulating recommendations are supported with effective actions, for example, making available server infrastructure, organising a workshop or setting up a working group.

3. To re-orientate. A re-orienting recommendation aims for a drastic reversal in an existing situation, but in the long term leads to the greatest results. It requires the most efforts on the part of policy and the most intensive discussions with the other actors in education.

6.0. Recommendation 0: ICT as an aid for individual self-development

This recommendation transcends the strict description of the tasks of the working group, in the sense that it is independent of whether or not Free Software is used. However, the working group notes that in too many places in current educational practice, the ICT efforts focus on short-term objectives: instead of the purely instrumental skills, the *real* goal of ICT should be to support pupils and teachers in their personal self development.

In concrete terms, the policy should therefore be inspired by the aim for efficient ICT, in which "efficiency" corresponds to an evolution through the following three stages:

1. Instrumental: the familiarisation with the "instrument computer", i.e. learning to use the mouse and the keyboard, starting the computer and applications, etc.

2. Useful: the *use* of the applications provided in the working environment or for the exchange of information with administrations; i.e., being able to deal with web forms, electronic banking, etc. These skills in the first place (and often in the only place) increase the efficiency of the *administrations* concerned, and not that of the individual user.

3. Self-development: *the ability to use ICT creatively* in order to increase *personal* efficiency and intellectual development; i.e., the computer gives the individual more opportunities to be constructive and creative; to gain more use from electronically available information; to be able to convert this information into personal expertise and into solutions for his own problems via freely available international networks between individuals with the same interests; to take advantage of more opportunities to develop at the professional, educational or recreational level; etc.

"Free Software" (in the broad sense of the term) is a very suitable "microcosm" to stimulate the above-mentioned evolution. The majority of the individual, voluntary contributions to Free Software come from people who are motivated by the unique opportunities for non-profit self-development provided by the Free Software community. Furthermore, Free Software is based on a strong modular structure and is (therefore) more than one alternative solution for almost all ICT needs. This leads to a better insight in the approach to the use of ICT and the development of competences related to the learning process.

6.1. Recommendation 1: Free Software as a model

Free Software can only have a place in school ICT *if* the teachers, ICT coordinators, heads and authorities understand what Free Software is, how they can make use of the competitive advantages of Free Software and what terminology and concepts are common to the Free Software microcosm. This objective is supported by the following recommendations:

- To stimulate. All the teachers should have an opportunity to be familiarised with the development model of Free Software and be stimulated to take part in this. The most important educational reason is that this model of constant and not-for-profit cooperation between all the interested parties can be applied to the creation of *all* non-material educational teaching materials. Therefore not only to software, but also to "ordinary" teaching materials, such as textbooks, exercise sheets, sound and visual fragments and illustrations.

- To facilitate. The familiarisation with the developmental model of Free Software can be accelerated by making available a few centrally managed servers for discussion forums, mailing lists, etc., so that the people involved can exchange information and experiences quickly and without restriction (and with interested "outsiders").

  It is important that Education takes the final responsibility for this server infrastructure, guarantees its neutrality and stimulates experts in institutes of higher education, universities and businesses to participate in the forums. It is *not* necessary for Education to act in a way that directs the discussions or the creation of educational Free Software: a certain degree of "self-regulating chaos" is a natural and necessary component of creative discussion and cooperation.

- To reorient. Although the government has the final responsibility for the server infrastructure, it can be appropriate to make use of Public Private Partnership (PPP) for its management and financing, at least as long as the supplier independence and the use of open standards remain guaranteed.

PPP, institutes of higher education, universities and teacher training colleges also provide natural cooperative ventures which can be used for the initial *creation* of new projects in accordance with the Free Software model, because this is the stage which is most intimidating for teachers.

6.2. Recommendation 2: ICT skills and "attainment targets"

In principle, "attainment targets" for ICT are essentially no different from those for other "subjects", in the sense that the pupils must have mastered a number of basic skills, must understand a number of basic concepts and must learn the attitude to apply these concepts correctly. However, in the field of ICT, there is a practical handicap in education: the technology is still relatively new and no one yet has a clear grasp of exactly which skills must be learned, how and at what age.

The following recommendations support a creative and interactive exploration of ICT skills and attainment targets:

- To stimulate. The Ministry of Education must continue to have confidence in the current initiatives of its own administration to support ICT because this administration has proved that it has a healthy, neutral and realistic vision of ICT in education, *and* because it has adopted the right tone to approach this vision with people in the field. This takes place, inter alia, with progressive but pragmatic projects such as ICT on the menu, and ICT competences in primary education, and with the introduction of simple but powerful conceptual structures such as the competence diamond.

  These initiatives must be further stimulated and extended (by the above-mentioned cooperative ventures) in the following directions:

- Entering into a debate with the people in the field, to identify all the really essential ICT skills and look for the best teaching methods to teach these skills to pupils and teachers.

- Creating teaching materials which will provide an insight into the various aspects of information processing: its creation, presentation, transformation, storage and exchange. Making information useful is one of the basic tasks of the ICT, but pupils and teachers are not given nearly enough explanations or clarifications about what information is, what its value is and how to deal with it.

- Emphasising the main agreements between ICT concepts on the one hand, and non-ICT concepts in everyday life on the other hand: almost all the tasks which are carried out by computer are actually no more than a modern way of doing "old" things in a (potentially) more efficient way. For example, a telephone directory is similar to a databank in many ways; the mail service is organised in a similar way to electronic mail; the security of computer data takes place with concepts which are centuries old; etc.

- Emphasising the importance of understanding ICT at the conceptual level. This applies at every level of education, i.e. certainly also in technical and vocational education. Pupils (and teachers) in these forms of education also benefit from an understanding of the different ICT concepts and skills.

- For example, a pupil doing a commercial course must understand the needs, possibilities and problems of electronic information exchange between companies, and the fact that open ICT standards play a key role in this; an operator of computer equipment should not lose his place in the labour market when the generation of machines for which he was trained is replaced by a new generation of computer support; an employee working in a secretariat must learn the enormous advantages of a databank in comparison with the much more traditional spreadsheets; etc.

    - To facilitate. The priority of the Ministry of Education should not be to attract *more people* into ICT support jobs, but (i) to provide the facilities for these people to join and play a leading role in the European networks developing for Free Software for education, and (ii) to make use of the expertise and manpower in institutes of higher education and universities. For example, in the search for the "correct" ICT skills; for the organisation of study days on a regular basis about the concrete technical and educational aspects of educational ICT projects in ICT Free Software; for the initiation of new educational projects, for example, on the basis of a thesis; for helping to develop teaching materials; etc.

    - One way of involving universities and institutes of higher education more actively in educational ICT projects in Free Software is to recognise their

contributions as real social services and to "pay" for these with the already existing systems of evaluation. For example, this is possible by charging for them in the same way as for scientific publications.

- To reorient. There are different "ICT tools" for each of the above-mentioned aspects of information; but none of these ICT tools is the best solution for all ICT tasks. All ICT courses should become supplier-independent, focusing on general product-neutral ICT skills.

6.3. Recommendation 3: Infrastructure

The most important contribution of the recent evolution of the internet for education is undoubtedly this: the rapid and unrestricted access to information has become more important than the traditional *possession of information*. Instead of acquiring all the information from the information sources which are available in the school library, modern pupils and teachers benefit more from knowing how they can consult the correct information sources on-line and where they can communicate with *people* who can provide clarification and structure for these information sources. (Obviously these on-line possibilities should be used *together with* the traditional libraries.)

Therefore, a familiarity with both technological and human networks, as well as the skills to make the best use of these are fundamental to survive in a "knowledge society". Therefore the policy must provide the infrastructure for this, which can be achieved with the following recommendations:

- To stimulate. Stimulate pupils, teachers, and school administrations to make increasing use of web-based ICT tools. Reports and projects can often be produced on a web page just as well as on paper. Administrative applications are often easier to control on strong centrally managed servers. Exercise materials for lessons can be offered with a web browser, etc.

  The traditional "personal computer" paradigm is only still appropriate for ICT tasks in which the data transport between the "client" and the "server" would require significantly more time than processing the data. For example, drawing programmes or photo manipulation.

- To facilitate. Pupils and teachers must have access to a larger number of "thin clients". Old PCs with Linux are the perfect solution for this. Thin clients often no longer even require a hard disk, ventilator or CD-Rom and are therefore becoming extremely maintenance friendly.

  Web-based teaching materials must be developed to familiarise pupils with the suitable ICT tools in a producer-neutral way.

- To reorient. ICT practice in education must evolve from the objective of *One PC per ten pupils* to the objective *One server network per cluster of schools*. This evolution is desirable because of the increased efficiency of ICT resources with the far-reaching use of thin clients and web-based applications (because the same number of people can keep a much larger number of computers ready for operation), as well as because of the increased quality of the ICT skills taught in this way. The success of the *GSM annex pocket computer* proves that society is already rapidly evolving from the "computer" as a clearly present, centralised and personal piece of equipment towards "invisible distributed systems" consisting of networked "computer" centres where users can have access to all their ICT information and applications anywhere, through a range of integrated software components of different suppliers.

  Free Software supports and respects open standards. Open standards are the only way in which ICT can evolve towards the aim of networked and producer-neutral access to information: the seamless and independent integration of a complex infrastructure of programmes, data and computers.

6.4. Recommendation 4: Teaching materials

ICT in education, with or without Free Software, can only be successful if teachers can make use of a large range of ready-made teaching materials which they can easily adapt to their own specific needs. Stimulating the creation of these teaching materials should be one of the priorities of the ICT policy.

- To stimulate. At the moment, many teachers are already motivated to create teaching materials. However, most are not familiar with the power and leverage effect of the Free Software model. The government should inform teachers about this and stimulate them to cooperate, not only to work on software but also on other educational materials with Free Software licences.

- To facilitate. Just as the insights into the "correct" ICT skills are still underdeveloped, the availability of supporting teaching materials is also still below par. The policy should establish actions to bring together as many actors as possible from the Education target groups and outside, in workshops and conferences, at the national and international level, to think about and experiment with Free Software teaching materials together.

- To reorient. In view of the limited current provisions of teaching materials related to and with ICT, both in Free Software and in closed software, there are few or no reorienting actions which can be undertaken.

6.5. Recommendation 5: Training

One important aim of “ICT at school” should be to familiarise pupils, teachers and administrations with the technology behind ICT. It is only in this way that they can respond critically to the constantly increasing commercial pressure of ICT companies: they learn to distinguish the important matters from secondary matters; they understand the useless superficialities of ICT products and what is essential; they are no longer overwhelmed by marketing slogans; they can compare the advantages and disadvantages of various ICT solutions on the basis of their technical characteristics.

Familiarity with ICT technology increases the quality of all teaching materials which can be produced by people in school practice. This aim can only be achieved if the teachers and ICT coordinators themselves have access to training and the related teaching materials.

- To stimulate. A proposal to make a *familiarity with ICT technology* one of the aims of the programme is often received with scepticism because it is considered to be too difficult: after all, we do not turn all our pupils into car mechanics do we?” However, understanding ICT technology is not difficult at all once the right connections are made with familiar centuries-old technologies in normal society: after all, ICT is usually no more than automating these already existing processes with a computer.

  Teachers should be stimulated to first of all discover these “de-mystifying” insights themselves, and then to translate them into the best educational ways for their own school environments. However, self study and support for training is required for this and Free Software licences are ideally suitable for all the related training materials.

- To facilitate. The policy must take action so that the teacher training courses devote attention to this training in the various universities and institutes of higher education, opting for Free Software licences in this respect in order to stimulate the large-scale dissemination and improvement of the materials as much as possible, and in this way keeping the development costs down.

- To reorient. Current school practice places too much emphasis on the WYSIWYG and desktop paradigms for dealing with ICT: the characteristics and complexity of computer and network technology are, on the one hand, abstracted behind icons and graphic tools, while the standard graphic presentation is that of the personal desktop of the user. This undoubtedly reduces the threshold for working with ICT, but at the same time waters down the insights in these real characteristics and complexities, particularly with regard to anything which concerns networks and distributed information access and processing. The result is that our pupils and teachers have little or no understanding of exactly what are the causes of ICT products which crash, or the fundamental differences between local and global information processing, or how they can achieve their ICT tasks in the technologically most efficient way.

> Therefore the policy must make efforts to place the alternatives for the WYSIWYG paradigm (back) on the ICT menu and to adapt the ICT courses in this sense.

The way in which the Free Software community deals with the technology and the way in which the community provides its "courses" itself, in practice prove to stimulate the self-sufficiency and self-confidence of users to a high degree. After all, Free Software projects do not need to create new buzzwords or to think of new names for old terms for purely commercial reasons. In contrast with closed software producers, Free Software projects are also always very open about *how* their technology works, and on the basis of which knowledge and concepts available elsewhere it is built up. Each of these projects devotes attention to the training of users and employees with on-line documentation, FAQ documents (Frequently Asked Questions) and news groups or mailing lists.

7. Decision

Free Software has long gone beyond the stage of large-scale technological experimentation and has become a permanent presence in a growing number of sectors. However, this has not yet been achieved in education. Therefore education policy must recognise the evolution of Free Software and make maximum use of it, eventually playing a stimulating and leading role. This is not only applies for the software itself, but for all the non-material educational projects, such as courses and teaching materials. Pragmatic but explicit support from the Ministry of Education will only increase the existing leverage effect of Free Software. One of the most important lessons which the other large "users" of Free Software outside education have learned is that Free Software cannot be directed or planned but can only be stimulated. Unless large budgets are provided for the creation of Free Software to order. However, even that does not provide a guarantee of success, in which "success" is defined as "accepted by the (international) school community as a project that is worth using and worth working on".

It is possible to achieve synergy between the objectives and actions of Education and the objectives and actions of private companies. The complete independence of the educational project of Education must be guaranteed in this.

> In short, there are no good reasons to postpone the introduction of Free Software in education any longer (in the long term it should become very far-reaching). In the first place, this should be achieved on the basis of encouragement and not on the basis of obligation or by blindly increasing the ICT budgets.

At the moment, the main advantages of Free Software are: the technological qualities for the *support of ICT infrastructure*, the *absence of licensing costs* and the *role model for the creation of educational added value*, both for software and for teaching materials. The greatest current obstacles are: the *lack of familiarity* of society with Free Software, the *resistance to change* and the *limited availability of teaching materials.*

The recommendations in this document will eventually lead to (i) a reduction in the costs of purchasing and managing software, (ii) a larger collection of teaching and training materials that can be freely used, and (iii) the increased quality and efficiency of ICT integrated in school practice.

Appendix 1: List of terms

BSD

Berkeley Software Distribution, one of the oldest and most widely used licences for Free Software.

Commons-based peer production

Commons-based peer production has been the basis for the creation of added value in a community of like-minded people for centuries; for example, in scientific research: everyone who has the qualities to evaluate, research and participate in it has the same access to the existing knowledge, and the same possibilities of questioning, expanding and applying this knowledge, etc.

Creative mass

Creative Commons is the most recent of the large “Free Software” communities, but with an emphasis on non-software creations: texts, music, illustrations, film, etc. The name comes from the principle of commons-based peer production.

Desktop

Most operating systems provide a graphic interface which is inspired by the familiar concept of the desktop. This was familiar to the generations which grew up with traditional desks full of paper folders, before electronic communication between computers was commonplace, and before the amount of information which the average computer user must or wants to use became much too extensive to fit on a desk. At the moment, the internet is becoming increasingly important in everyday ICT practice and therefore the desktop paradigm is ready to be renewed.

Extegration

The integration of different software modules from the same manufacturer in such a way that similar modules of competitors cannot be integrated. This automatically leads to lock-in.

Fair use

The details of the legislation on copyright often differ from country to country. However, there is a form of “fair use” of materials protected by copyright everywhere. For example, it is permissible to cite fragments from literary or musical works. It is rarely clear how large these fragments may be, or how often and in what context they may be used, and this is finally determined by the interpretation of a court.

FLOSS

Free Libre Open Source Software, an English compromise name for Free Software, mainly used in texts which want to remain as neutral as possible. After all, in the “Free Software community” there is quite a lot of sensitivity related to the exact name.

Free Software

The term Free Software is also known as Open Source software.

This document uses the term in a way which goes beyond merely the software, in particular, (i) all the immaterial knowledge and skills which can be produced, distributed and taught with computer and communication technology, and (ii) a mentality and attitude which stimulate self-sufficiency, active, and non-profit cooperation and lifelong learning.

Free use

Free Software can be used, adapted and integrated with other software free of charge on an unlimited number of computers. This concept not only applies for the software, but for all the immaterial forms of knowledge and information, such as documentation, sound, image, and figures.

Freeware

Software made available by the authors free of charge, *without* the source code of the software and without a licence which makes it possible to use the software freely as Free Software.

Closed software

The opposite of Free Software: the software may not be copied in an unlimited way and the source code is not available. This text sometimes uses “closed software” as a synonym for *commercial software*, although in fact this is too narrow an interpretation of the term “commercial”: after all, there is nothing to stop the commercial exploitation of Free Software as long as the licence conditions are met.

GPL

General Public License, the most popular licence for Free Software. This licence was established in 1985 and has proved to be surprisingly strong for the legal protection of the efforts and interests of all those involved, even in the era of the internet which was only a distant dream in 1985.

The strength of the licence mainly lies in the condition that all the programmes that are combined with a GPL programme must also have the GPL licence themselves (in this context also see the LGPL licence).

GPL-compatible

The GPL licence does not allow the software which is distributed under licence to be integrated with software with any other licence. More information about which licences can be used without problems can be found on the website of the Free Software Foundation and the Open Source Initiative.

Intellectual property

This is one of the popular *newspeak* terms from the commercial software world, which would like to persuade the general public that the creation and distribution of knowledge should be protected in completely the same way as material “inventions”, for example, by patents. However, this attitude completely ignores a number of fundamental differences between knowledge and material things: knowledge grows when it is shared; a system of patents has degenerated worldwide by an inflation in the number of patents granted for concepts and developments which are much too trivial; the advantages of protecting knowledge are privatised by the system of patents and therefore rarely lead to an increase in the *macro-economic* use which was the original intention of patents.

Kernel

The “kernel” of an operating system, for example, Linux or FreeBSD. Free Software operating systems usually follow the UNIX model, in which the kernel is kept as limited as possible and in which other general infrastructure programmes, such as the graphic interface, are completely separate components. However, in its Windows operating systems, Microsoft has extegrated many of these components, and this competitive monopoly advantage has been used to force most of its competitors off the market.

LGPL

Lesser General Public License, the most popular licence for Free Software apart from GPL, particularly for support software. The difference with the GPL is that software with a LPGL licence can be combined with software with other licences in an unlimited way.

Licence

Software is protected by copyright automatically and without the author having to take any action, in the same way as music, written texts, photographs etc. This means that the free use of software is not permitted. However, the author of the software can change this standard protection by explicitly adding a user's licence to his creation, which determines what users are permitted to do with the software. Commercial providers are usually extremely restrictive with these licences, while on the other hand, the makers of Free Software are extremely permissive.

Linux distribution

There are thousands of Free Software programmes, which can all be downloaded from the internet and tested. However, in this way, there is a great deal of work involved for an individual or organisation to collect and integrate the correct group of software. Therefore a number of commercial and non-commercial organisations have developed which sell this job of collation as a service. The result is a *distribution:* a collection of tested, documented and coordinated programmes. The most widely distributed distributions are Debian, SUSE, Mandrake, and Red Hat.

Live CD

A CD-Rom which can be used to start a PC. Many Free Software projects make these CD-ROMs available, (particularly) to familiarise Windows users with their software in an easy way.

Lock-in

Most closed software producers follow the strategy (i) to offer their products as very large monolithic blocks, and (ii) to achieve the storage of data in a secret producer-specific format. For example: you have to buy the whole of Microsoft Office, because you cannot buy *Microsoft Excel* or *Microsoft Word* separately; the information which you enter in these programmes is automatically stored in files for which only Microsoft knows the structure.

These aspects make the exchange of data or the replacement of part of the function with provisions of other producers difficult or impossible. This means that the client of those products is increasingly dependent on one producer because in practice, transferring to or making use of the software of other providers costs a great deal of time and money. This supplier dependency is known as *(vendor) lock-in*.

Meritocratic

Only based on inherent qualities. Virtually all the Free Software projects evaluate suggestions and contributions by "outside" employees on the basis of their quality and not on the basis of the status or social position of the provider.

Object-oriented ICT

Real ICT skills go far beyond the procedural ICT which is commonplace in current education practice. Pupils have to learn the *ICT objects*, i.e. the combination of data, storage and communication possibilities and the operations which turn the data into real *information.* This is all completely independent of specific software programmes which are available for ICT objects. The object-oriented ICT approach is not as strange as one might be inclined to think at first sight. On the contrary, it is the obvious approach in all traditional non-ICT lessons.

Open Standard

Computer users wish to exchange information between different programmes and different computers. This means that both the "sender" and the "receiver" must present the data in the same way. The description of this communication protocol is a *standard*. ICT systems can be much better integrated if this standard is *open*, i.e., completely documented, available to everyone, and free of the arbitrary conditions of a single producer. See openstandaarden.be for a more detailed explanation.

Operating system

The software which operates the hardware of a computer and provides a programming interface with which the application programmes can be made, without the need for the programmers to know on which computer their software will be used. Microsoft Windows, and Apple Mac OSX are two examples of operating systems. Linux is not a complete operating system, but only the kernel.

Peer

A person of the same class, rank, expertise, etc. as others in a particular community. *Peer review* is the age-old principle for evaluating the quality of a (scientific, technical, cultural…) creation (only) by people of a similar scientific, technical or cultural level. This principle has been fundamental for progress in science and technology in the western world since the Renaissance, and much longer for art and culture.

Platform

A (computer) platform is roughly, a specific combination of a processor, peripheral equipment, communication box, and operating system. Examples of popular platforms are (i) *Microsoft Windows* on Intel processors and PCI box, (ii) *Mac OSX on Power PC* processors and PCI box, (iii) *Linux* on dozens of different sorts of processors and boxes, and (iv) *Linux* on *clusters* and *grids*.

Platform independent

Software which can work on several platforms. The more platforms are supported, the more chances of a fair competitive ICT market and the less lock-in.

Procedural ICT

The great majority of ICT courses work in a very shortsighted way: they are based on a *single specific programme* (for example, Word or FrontPage) and then they explain with which menus and mouse clicks it is possible to achieve a *particular result*, for example, changing a piece of text into bold print). This means that the trainee is concerned mainly with the *procedure* and does not acquire sufficient long-term *skills*. The examples given above are concerned with techniques which centuries of experience in the art of printing have taught us, i.e. that the *content* of a document should be supported as well and as transparently as possible by the *design* of the document.
Also see object-oriented ICT.

Shareware

Like freeware, in the sense that a free version of software which is not free of charge can be used, usually with the difference that there may be a fee for shareware, possibly after a trial period that is free of charge.

Software in the public domain

This software can be used freely and the author even surrenders the complete copyright to the users. This licence is used for example, by a number of government institutions in the United States, based on the argument that the taxpayer only has to pay once for the use of software which is made by a government department.

Source code

The codes written and read by people in which the programmes are made. The availability of the source code is essential to adapt and extend the programmes.

Terminal

The general breakthrough of the *Personal Computer* has led to a conviction in the majority of ICT users that you need a computer with a large memory and a large hard disk with all the software before you can do anything useful with it. This is not true at all, and in the UNIX/Linux world, the concept of a *terminal* has always been used: this terminal only has very limited possibilities (often not even a hard disk), because it is no more than a keyboard and a screen which provides the interaction with a programme which runs on a *server*. This server is a substantial computer as it operates dozens and often hundreds of terminals. *Terminal Servers* are once again becoming popular for two reasons: (i) networks have become extremely rapid, and (ii) it is much simpler to maintain a number of terminals (“thin clients”) and one server, than dozens of “fat clients”.

WYSIWYG

What You See is What You Get, an abbreviation which is often used to indicate programmes which are entirely operated graphically. In practice, this regularly leads to procedural ICT use, which is a significant obstacle for efficient ICT.

Appendix 2: References

Debian

The most large-scale Linux distribution which is completely independent of commercial partners.

ICT on the menu

65 recipes for the use of computers in primary school. This is a publication by the Government of Flanders which introduces and explains 65 freeware programmes, in specific educational contexts in primary education. The programmes are available on the website of Klascement.
This publication shares the very useful *ICT competence diagram* with another publication of the Flemish Department for Educational Development, viz. the ICT competences in primary education.

ICT competences in primary education

http://www.ond.vlaanderen.be/dvo/basisonderwijs/ICT/indexict.htm

European Schoolnet

http://www.eun.org/portal/index-nl.cfm

Free Software Foundation

http://www.fsf.org.

Open Source and Standards in education

http://www.ossinhetonderwijs.nl.

Open Source Initiative

http://opensource.org/.

OpenStandaarden.be

http://www.openstandaarden.be

UNIX

The collective name for a number of operating systems based on very successful developments at Berkeley University and Bell Labs. Some versions were continued as Free Software (for example, FreeBSD) while companies such as Sun, IBM, and Hewlett-Packard developed their own commercial version. A UNIX system consists of a large number of modules which can be very flexibly integrated in powerful programmes.

All the commercial UNIX suppliers made the historical error of only sharing their own new developments to a very limited extent. The result was a growing problem with regard to the exchange of different UNIX systems, and these divisions made them an easy prey for Microsoft Windows.

Appendix 3: Authors

Herman Bruyninckx

The chief lecturer at the Department of Mechanical Engineering at the Faculty of Applied Sciences, Catholic University of Leuven. Since 1998 he has been an active advocate of Free Software in education, industry and government. His academic research (robotics, software for robot and machine operation) makes extensive use of Free Software. He is involved in federal and regional governments as an advisor, and in the *Association of Flemish Towns and Municipalities.*

Mark De Quidt

ICT coordinator and teacher of Applied Information Technology at the Royal high school in Antwerp. He went into education after a career with IBM.

Kim Lauwers

An advisor for the Media innovation division of the Department of Science, Innovation and the Media, Ministry of the Flemish Community. An industrial engineer with nine years of work experience in information technology in several ICT companies. The author of several contributions on Free Software for the Eflanders newsletter.

Eric Verhulst

The head of the ICT department at the Plantijnhogeschool in Antwerp. A member of the steering group and the editorial team of the “Programmamatrix”. An employee at REN Antwerp, with a course on an OSS electronic learning environment, amongst other things. Member of several steering and working groups for higher education. Designer and co-author of a school administration package used in about 300 institutions from nursery to higher education. A career as a teacher in primary, secondary, higher and adult education.

Wilfied Feijens

Employee of Anywize, Cisco Academy and other ICT training initiatives.

URL: http://people.mech.kuleuven.ac.be/~bruyninc/oss/oss-onderwijs.html