

EFI-CH

Rapport final

Suisse romande

—

_oci

Option complémentaire Informatique au gymnase

Sommaire

— Introduction	4
Cadre général	5
L'informatique dans les gymnases suisses : une histoire mouvementée	5
Comparaison internationale	5
EFI-CH : une coopération unique entre plusieurs hautes écoles	7
EFI-Deutschschweiz	8
OCI en Suisse romande	8
OCI-Ticino	8
Offres et participants	8
Conception de la formation OCI en Suisse romande	10
Module de base	11
Modules d'approfondissement	11
Semaine Séminaire (cours bloc)	13
Semaine de stage	13
Travail de Diplôme	13
OCI : mise en œuvre du curriculum	14
Exemples de pratiques d'excellence	15
Thématique 1 : Apprendre à programmer dans le monde du multimédia	15
Thématique 2 : Pas de pratique sans théorie	19
Thématique 3 : D'accord pour suivre les tendances en informatique, mais pas pour suivre la mode	23
Exemples de pratiques d'excellence dans le domaine de la didactique de la discipline	26
Idées fondamentales	26
Great Principles of Computing	28
OCI : évaluation des cours en Suisse romande	29
Evaluation du module de base	30
Evaluation des modules d'approfondissement	30
Evaluation de la semaine de séminaire (cours bloc)	31
En résumé	31
OCI : conclusion	32
Perspectives	33

Avant-Propos

Le présent rapport se fonde sur le rapport analogue rédigé pour la Suisse alémanique. Ce dernier rend compte des différents projets partiels dans les régions linguistiques, ainsi que du projet EFI-CH dans son ensemble. La désignation EFI-CH fait référence au projet national, tandis que « EFI-Deutschschweiz » désigne le projet partiel pour la Suisse alémanique et « OCI » (option complémentaire Informatique), le projet partiel pour la Suisse romande.

Le projet partiel pour la Suisse romande n'impliquant que des professeurs de sexe masculin, seule la forme masculine est employée dans le texte du rapport. Bien que la formation continue ait été suivie par des femmes et des hommes, nous avons également choisi le masculin générique pour désigner les participants à ces cours. Les énoncés et les conclusions du rapport s'appliquent aux deux sexes.

Introduction

Dans le présent rapport, nous dressons le bilan de l'offre de formation continue pour les enseignants de gymnase (OCI). De 2008 à 2012, une centaine d'enseignants ont suivi des études d'informatique en cours d'emploi, afin d'obtenir les qualifications nécessaires pour l'enseignement de l'option complémentaire Informatique. Le document présente les conditions cadres de la formation, le programme, quelques exemples de modules réalisés pendant les cours et des statistiques sur la mise en place de l'option complémentaire dans les gymnases suisses. Le rapport a pour but d'informer les personnes directement impliquées, les principales parties prenantes et les responsables de la formation sur le déroulement de ce projet. En outre, il résume les principales constatations pouvant servir de base à des manifestations futures éventuelles.

Le projet OCI a uniquement pu être réalisé grâce au soutien généreux de la Fondation Hasler, qui a financé l'élaboration du curriculum et l'organisation des modules de formation dans les différentes hautes écoles. Elle a également mis à la disposition des cantons les moyens nécessaires pour la décharge des enseignants qui ont souhaité suivre cette formation. Au total, la Fondation Hasler a investi cinq millions de francs suisses dans le projet EFI-CH (ensemble de la Suisse), jusqu'ici le projet le plus important du programme « FIT – Fit in Informatik in der Bildung ». Conçu pour une durée de dix ans et doté d'un budget total de 20 millions de francs suisses, FIT a pour but d'inscrire l'informatique dans la culture générale.

Helmar Burkhart, Université de Bâle

Martin Guggisberg, Université de Bâle

Beate Kuhnt, Université de Zurich

Martin Lehmann, Pädagogische Hochschule Bern

Jacqueline Peter, Centre suisse de formation continue des professeurs de l'enseignement secondaire (WBZ CPS)

Roger Mauron, responsable du projet partiel Suisse romande

Lausanne, en mars 2012

Cadre général

Lors de la révision partielle, en 2007, du Règlement de 1995 sur la reconnaissance des certificats de maturité gymnasiale (RRM95), l'informatique a été inscrite dans la liste des options complémentaires. Depuis, elle fait partie intégrante du catalogue des branches proposées au gymnase. L'option complémentaire, l'option spécifique et le travail de maturité sont choisis individuellement par chaque élève. Les conditions cadres relatives à l'enseignement des options complémentaires varient fortement d'un canton à l'autre : la dotation horaire oscille entre 2 et 6 leçons hebdomadaires sur une base annuelle ; tantôt l'option complémentaire est enseignée au cours de la deuxième, tantôt pendant la troisième ou la quatrième année du gymnase ; dans certains cas, l'enseignement s'étale sur une année, dans d'autres sur les deux dernières années. Malgré ces différences, toutes les options complémentaires ont un point commun : elles comptent pour la promotion ; à ce titre, la note obtenue figure dans le certificat de maturité. Afin que l'informatique puisse s'établir définitivement comme 14e option complémentaire – qu'elle soit choisie par les élèves –, la conception du cours, d'une part, et des connaissances de l'enseignant, d'autre part, devront être d'excellente qualité.

L'introduction de l'informatique comme option complémentaire a fait apparaître un besoin accru d'enseignants en informatique possédant les qualifications nécessaires. En effet, le RRM95 exige que les disciplines de maturité soient enseignées par des personnes titulaires d'un diplôme académique dans la matière en question. De plus, pour enseigner l'informatique, – et ce ne sont pas que les spécialistes qui le diront –, il est impératif de posséder des connaissances approfondies de la science éponyme. Il était donc urgent de prendre des mesures. C'est dans ce contexte que la Fondation Hasler a lancé le projet EFI-CH, portant sur la qualification des enseignants des gymnases suisses intéressés par l'informatique, mais ne possédant pas de diplôme dans cette discipline. En Suisse alémanique, le projet était placé sous la houlette de l'Institut d'informatique de l'université de Zurich, qui a élaboré une formation complémentaire en coopération avec d'autres hautes écoles, tandis qu'en Suisse romande, le projet a été porté par l'EPFL.

L'informatique dans les gymnases suisses : une histoire mouvementée

La discipline « informatique » au gymnase a connu une histoire mouvementée. Il est utile d'en rappeler les grandes lignes, afin de mieux comprendre la situation actuelle. Entre 1985 et 1995, l'informatique faisait partie du plan d'études gymnasial. Dotée de 2 à 4 leçons hebdomadaires sur une base annuelle, la branche était obligatoire pour tous les élèves. Les contenus de la discipline n'avaient pas été définis au niveau fédéral ou cantonal, mais dans les années 1980, l'accent était clairement mis sur la technique de programmation.

Conséquence de la généralisation des ordinateurs (PC) dans les années 1990, l'importance des programmes applicatifs dans les cours d'informatique n'a cessé de croître, de sorte que l'enseignement de l'informatique s'est résolument orienté vers les TIC. Parallèlement, la stratégie de l'informatique intégrée a fait son chemin, avec l'idée d'utiliser l'ordinateur et l'Internet dans toutes les disciplines. Ainsi, les élèves étaient invités à écrire leur rédaction d'allemand dans un traitement de texte ou à analyser leurs expériences de chimie au moyen d'un tableur. Les experts constatent aujourd'hui que cette approche intégrative est un échec, car elle exige que les enseignants eux-mêmes disposent de solides connaissances informatiques.

Pour l'enseignement de l'informatique proprement dit, la généralisation des ordinateurs et l'approche de l'informatique intégrée ont eu des conséquences néfastes : le cours d'informatique avait plus ou moins disparu du programme. L'introduction de l'option complémentaire Informatique permet à la discipline scientifique informatique de retrouver sa place au gymnase. De plus, elle contribue à combattre l'idée largement répandue que l'informatique se résume à l'utilisation d'un ordinateur.

Comparaison internationale

Une étude financée par la Fondation Hasler et réalisée par l'Institut d'informatique de l'Université de Zurich constate, qu'à l'instar de la Suisse, plusieurs pays européens, américains et asiatiques souhaitent ancrer l'informatique à titre de discipline scientifique dans leurs programmes scolaires. Elle montre également que la plupart de ces pays connaissent des problèmes similaires. Ces difficultés sont présentées en bref dans ce chapitre et décrites en détail dans le document¹.

¹ Guerra, Vania : Computer Science World Wide, étude réalisée à l'Institut d'informatique de l'université de Zurich, publication en avril 2012.

Dans la majorité des pays, il n'y a pas de distinction explicite entre l'utilisation des applications informatiques courantes, tels que les tableurs, le traitement de texte, les diaporamas et les moteurs de recherche, et les aspects scientifiques de l'informatique. Par conséquent, les cours d'introduction à l'utilisation de l'ordinateur, appelés cours TIC dans le présent rapport, sont assimilés à l'informatique. Les pionniers de l'introduction de l'informatique à l'école ont fort à faire pour lever cette confusion.

De même, la restructuration des cours TIC et des cours d'informatique est en cours dans de nombreux pays. A cet égard, il est intéressant de constater que certains pays de l'est de l'Europe passent de l'informatique aux TIC intégrées, tandis que les Etats-Unis et l'Europe occidentale entendent instaurer l'informatique comme discipline à part entière et introduire une formation de base en informatique de l'école enfantine jusqu'à la maturité. Dans les pays qui proposent une maturité avec spécialisation, l'informatique fait partie des branches obligatoires dans les sections scientifiques, comme par exemple en Israël.

En règle générale, la formation des enseignants est fortement axée sur les contenus pédagogiques et une réflexion critique sur le rôle de l'ordinateur dans la société. Le cours TIC est donc confié à des spécialistes de la pédagogie des médias, plutôt qu'à des spécialistes des applications TIC et encore moins à des enseignants d'informatique. Au niveau gymnasial, ces derniers se font encore rares. Cela n'a rien d'étonnant quand on sait que l'informatique en tant que science est une branche obligatoire à part entière dans un petit nombre de pays seulement. Dans la plupart des cas, les éventuels contenus obligatoires sont intégrés au cours de mathématiques.

En comparaison internationale, la Suisse se situe dans la bonne moyenne après l'introduction de la branche à option si l'on considère l'importance accordée à l'informatique. En revanche, en ce qui concerne l'introduction et la thématisation de l'ordinateur à l'école obligatoire et, en partie, au gymnase, notre pays se place plutôt en queue du classement. Toutefois, dans ce domaine également, les différences entre les cantons sont importantes.

EFI-CH : une coopération unique entre plusieurs hautes écoles

Le projet OCI-CH est le fruit d'une collaboration inédite entre plusieurs hautes écoles suisses : les professeurs des instituts d'informatique de treize hautes écoles au total ont coopéré pour élaborer un curriculum permettant aux enseignants d'obtenir une qualification dans une branche spécifique en cours d'emploi.

Les personnes concernées savaient d'emblée que l'enjeu était bien plus vaste que la formation en informatique des enseignants. Elles entendaient avant tout contribuer à la réintroduction de l'informatique au gymnase en assurant un enseignement de qualité de cette discipline. Cette motivation a permis l'élaboration, en peu de temps, d'un curriculum et le lancement de la filière après seulement un an de préparation. Les objectifs liés à cet effort étaient ambitieux ; ils n'ont d'ailleurs pas encore été entièrement réalisés :

- Former en informatique scientifique un nombre suffisant d'enseignants motivés pour enseigner l'option complémentaire au gymnase et favoriser ainsi l'introduction de cette option dans tous les gymnases de Suisse.
- Montrer aux élèves que la discipline scientifique informatique et l'utilisation d'un ordinateur sont deux choses fondamentalement différentes.
- Enseigner l'informatique comme un bien culturel et partant, comme base de notre société de l'information.

Pour des raisons organisationnelles, les formations n'ont pas pu être réalisées comme formation initiale ; elles sont sanctionnées par un titre CAS (12 crédits ECTS²), DAS (30 crédits ECTS) ou MAS (60 crédits ECTS), assorti d'une formation en didactique de la discipline de 10 crédits ECTS supplémentaires. La filière MAS a seulement pu être proposée en Suisse alémanique.

Trois projets partiels ont été mis en place, correspondant aux trois régions linguistiques. Les projets partiels de la Suisse alémanique, de la Suisse romande et du Tessin sont présentés en bref dans ce chapitre. Les explications fournies au chapitre suivant s'appliquent essentiellement au projet partiel de la Suisse romande (OCI).

EFI-Deutschschweiz

Les partenaires ci-dessous ont participé au projet partiel pour la Suisse alémanique :

Université de Zurich, Institut für Informatik (lead) ; Université de Bâle, Departement für Informatik ; Université de Fribourg, Departement für Informatik ; Hochschule Luzern, Forschung und Entwicklung in Informatik ; Pädagogische Hochschule Zentralschweiz Luzern, Zentrum Medienbildung ; Pädagogische Hochschule Bern, Institut Sekundarstufe II.

OCI en Suisse romande

Plusieurs représentants de l'EPFL ont assisté aux premières discussions en Suisse alémanique sur l'élaboration d'une formation complémentaire et d'un curriculum. Ils ont aussitôt relevé le défi et ouvert, mi-2008 déjà, une filière CAS à l'EPFL. Les deux éditions du cours ont été suivies par 36 enseignants au total. Dans un deuxième temps, un approfondissement des connaissances acquises au cours CAS a été envisagé. Dès le semestre d'automne 2009, l'EPFL a proposé des modules d'approfondissement, qui ont permis aux titulaires d'un CAS de compléter leur formation par un DAS. Les modules d'approfondissement ont également été organisés pour deux volées. Au total, 17 enseignants ont obtenu un diplôme DAS de 30 crédits ECTS. Quatre hautes écoles ont collaboré pour élaborer et organiser les modules d'approfondissement :

Ecole polytechnique fédérale de Lausanne (EPFL), School for Computer and Communication Science (lead) ; Université de Fribourg, Département d'informatique ; Université de Genève, Département d'informatique ; Université de Neuchâtel, Institut d'informatique.

OCI-Ticino

Deux représentants de l'Università della Svizzera italiana (USI) ont également participé aux entretiens préparatoires de l'avant-projet. Sur la base du curriculum élaboré, une filière DAS a été créée au Tessin en collaboration avec la Scuola universitaria professionale della Svizzera italiana (USPSI). Les cours ont été proposés dès mi-2009. Contrairement à la Suisse romande, où aucun cours de didactique de la discipline n'a pu être organisé faute de spécialistes francophones, le Tessin a mis sur pied une coopération avec la didactique de la discipline Informatique de l'EPFZ. Neuf participants issus des cinq gymnases du canton suivent actuellement la filière DAS, qui leur permet d'obtenir 30 crédits ETCS en informatique et 10 crédits en didactique de la discipline. Les cours prendront fin en 2012. Le sous-projet est réalisé en coopération avec les hautes écoles suivantes :

Università della Svizzera italiana (lead) (USI), Facoltà di scienze informatiche ; Scuola universitaria professionale della Svizzera italiana (USPSI), Dipartimento tecnologia innovative ; ETH Zurich, Departement Informatik.

Offres et participants

Plusieurs générations d'enseignants, âgés de 30 à 56 ans, ont manifesté leur intérêt pour la formation : certains avaient déjà enseigné l'informatique au gymnase au début des années 1990, d'autres avaient acquis une vaste expérience de l'informatique pendant leurs loisirs, d'autres encore enseignaient les sciences humaines, mais étaient intéressés par l'informatique. Par ailleurs, les notes obtenues dans les travaux écrits ne correspondaient pas toujours au bagage académique des participants : tel enseignant de lettres pouvait très bien faire une bonne note dans une matière plus proche des sciences naturelles. Au final, les critères déterminants pour réussir la formation complémentaire étaient la curiosité, l'intérêt et la persévérance.

Le choix de la filière (CAS, DAS ou MAS) était basé sur les exigences des cantons et les acquis des participants. Une attention particulière a été accordée à ces facteurs lors de la sélection des participants, l'objectif pour la Suisse romande étant de réunir un maximum d'enseignants dans la filière diplôme (DAS).

Les enseignants romands pouvaient opter soit pour la filière certificat (CAS), soit pour la filière diplôme

(cf. ci-dessus). Ils ont été sélectionnés en fonction de leurs qualifications antérieures et de leurs motivations. Mi-2012, 106 participants au total, provenant de trois régions linguistiques, auront terminé la formation continue ou complémentaire. Le tableau 1 montre la répartition des titres obtenus. Les participants représentent 64 des 127 gymnases cantonaux (cf. tableau 2).

Le grand nombre de titres obtenus montre qu'en l'espace de quelques années, quantité d'enseignants motivés ont suivi une formation complémentaire en informatique.

Ces personnes disposent désormais de la qualification spécialisée nécessaire pour enseigner l'option complémentaire Informatique. Ils contribuent ainsi à réduire la pénurie d'enseignants spécialisés dans la discipline. Toutefois, à l'avenir, les enseignants d'informatique devraient être titulaires d'un diplôme universitaire en informatique.

	CAS	DAS	MAS	Total
Suisse alémanique	20	4	33	57
Suisse romande	20	20	–	40
Tessin	–	9	–	9
Total	40	33	33	106

Tableau 1 : Nombre d'enseignants dans les différentes filières de formation continue

Programme EFI-CH	Canton	Nombre (part.)	Titre (part.)
EFI-CH Deutschschweiz	AG	1	MAS (1)
	AR	1	MAS (1)
	BE	9	MAS (1) ; CAS (8)
	BL	2	MAS (1) ; CAS (1)
	BS	3	MAS (3)
	GR	4	MAS (4) ; CAS (1)
	LU	11	MAS (3) ; DAS (4) ; CAS (4)
	NW	1	MAS (1)
	SG	5	MAS (3) ; CAS (2)
	SH	1	MAS (1)
	SO	3	MAS (2) ; CAS (1)
	SZ	2	MAS (1) ; CAS (1)
	VS	1	MAS (1)
	ZG	3	MAS (2) ; CAS (1)
ZH	10	MAS (9) ; CAS (1)	
OCI-Romandie	FR	4	DAS (1) ; CAS (3)
	BE	4	CAS (4)
	NE	7	DAS (2) ; CAS (5)
	GE	2	DAS (1)
	VD	18	DAS (8) ; CAS (5)
OCI-Ticino	VS	1	DAS (1)
	TI	9	DAS (9)

Tableau 2: EFI-CH : Cantons impliqués et nombre de participants

² ECTS European Credit Transfer System

Conception de la formation OCl en Suisse romande

Le programme de formation comprenait :

- 1 module de base (10 ECTS)
- 5 modules d'approfondissement (11 ECTS)
- 1 Semaine Séminaire (cours bloc à 1.5 ECTS)
- 1 Semaine de stage (1.5 ECTS)
- Travail de Diplôme (6 ECTS)

Le cycle complet représente environ 900 heures de travail (1 ECTS = 30 heures), réparties sur 18 mois. La moitié de ces heures sont données en présentiel, le reste est consacré au travail personnel.

Conçu comme une formation en cours d'emploi, ce programme n'exige qu'un jour de présence par semaine. Il s'adressait aux enseignantes et aux enseignants actifs du degré secondaire II des cantons romands ainsi que des régions francophones des cantons de Berne, Fribourg et Valais.

Travail de diplôme (6 ECTS)						
DAS MA 1 (2 ECTS)	DAS MA 2 (3 ECTS)	DAS MA 3 (2 ECTS)	DAS MA 4 (2 ECTS)	DAS MA 5 (2 ECTS)	Stage (1.5 ECTS)	Cours bloc (1.5 ECTS)
DAS MB (10 ECTS)						

Module de base

Le module de base consiste en une initiation à l'informatique et aux techniques élémentaires de programmation. Les modules d'approfondissement élargissent l'horizon en mettant l'accent sur les concepts fondamentaux de l'informatique. La liste ci-dessous illustre en bref les contenus de ce module CAS :

- **HTML**
 - Commandes principales de HTML
 - Création d'une page simple
- **Programmation**
 - Introduction à la programmation au moyen de JavaScript
 - Gestion des événements, entrée des données.
- **Graphisme**
 - Librairies graphiques supportées par JavaScript
 - Création de quelques applications graphiques
- **Bases de données**
 - Structures des tables de bases de données
 - SQL
 - Accès de bases de données au moyen de JavaScript
- **Objets**
 - Concept d'objet, héritage, structuration d'applications
- **XML**
 - Langage XML, utilisation d'arbres XML à partir des commandes AJAX
- **Parallélisme**
 - Concept du parallélisme
 - Objets actifs (processus)
- **Automates**
 - Concept d'automate, expressions régulières
 - Utilisation d'outils permettant l'étude des automates
- **Architecture des systèmes informatiques**
 - Architecture d'un processeur, langage assembleur, compilateur (basé sur un outil de génération de compilateurs), éléments de systèmes d'exploitation et de réseaux.

Modules d'approfondissement

Les cinq modules spécialisés ont été organisés et réalisés par des professeurs des universités de Fribourg, de Neuchâtel et de Genève ainsi que de l'EPF de Lausanne. La récapitulation ci-après présente le contenu des différents modules :

- **DAS-MA1 : bases de données (UNIFR)**
 - Modèle entité-association
 - Formes normales
 - Passage du modèle E-A au schéma de base de données relationnelle
 - SQL comme langage de manipulation de données
 - SQL comme langage de définition de données
 - Introduction à l'algèbre relationnelle
 - Interfaçage avec des applications web
- **DAS-MA2 : algorithmes & programmation (UNIFR et UNIGE)**
 - Notion de programme, instructions, exécution pas à pas, itérations, définition de méthodes (procédures), invocations
 - Variables et expressions, fonctions, affectation, typage, variables globales et notion d'état
 - Structures de contrôle
 - Programmation orientée objets
 - Entrées-sorties, fichiers de données, fichiers texte, communication réseau
 - Structures de données
 - Principe de la récursivité, application aux fractales, évaluation d'une expression, algorithme de tri rapide
 - Interaction avec l'utilisateur, composants et widgets, layout manager, gestion d'événements, application
- **DAS-MA3 : informatique théorique (UNIGE)**
 - Première partie : langages formels
 - Langages réguliers
 - Automates à états finis
 - Expressions et grammaires régulières
 - Langages hors contexte
 - Grammaires
 - Automates à pile déterministes et non déterministes
 - Langages récursivement énumérables
 - Machine de Turing
 - Logiques de 1er ordre

- Deuxième partie : calculabilité et complexité
 - Calculabilité effective
 - Hypothèse de Church et machines universelles
 - Langages récursifs et récursivement énumérables
 - Machines de Turing déterministes et non déterministes
 - Transformations polynomiales
 - Problèmes NP-complets et NP-difficiles
- Troisième partie : modélisation formelle
 - Système de transitions
 - Réseaux de Petri
 - Introduction à la logique et aux preuves
- **DAS-MA4 : systèmes informatiques (EPFL)**
 - Architecture des ordinateurs : composants de base d'un ordinateur, architecture au niveau d'un répertoire d'instructions, entrées-sorties et interruptions.
 - Systèmes d'exploitation : notion de processus et programmation concurrente, exclusion mutuelle, synchronisation en Java, gestion de la mémoire principale et pagination, concept de machine virtuelle.
 - Réseaux informatique et sécurité : nommage de l'Internet, les protocoles (UDP, TCP, HTTP), les menaces (spam, phishing, virus), les mesures de protection (firewalls, proxys, antivirus), les protocoles sécurisés (HTTPS, SSH).
- **DAS-MA5 : multimédia & visualisation (UNINE)**
 - Traitement (automatique) de la langue naturelle
 - Problèmes, questions et applications du traitement de la langue naturelle
 - Comptage statistique (loi de Zipf), modèle de langue et applications à l'analyse de corpus
 - Classification automatique (méthode de Naïve Bayes)
 - Principes de la recherche d'information
 - Traitement vidéo et audio
 - Ce cours traite des concepts et méthodes des systèmes multimédias..
 - Recours à Java Media Framework (JMF) pour la programmation.
 - Traitement des images
 - Ce cours présente le domaine du traitement de l'image et introduit les participants aux concepts et techniques de base.

Semaine Séminaire (cours bloc)

La didactique de la discipline « informatique » n'étant pas proposée en Suisse romande, le comité de pilotage (COPIL) avait inclus dans chaque cycle de cours une semaine de séminaire (cours bloc), consacrée surtout aux questions pratiques. Ce cours bloc avait pour but secondaire de favoriser le travail en groupe, puisque tous les participants étaient réunis dans un hôtel pour une semaine de travail intensif.

Semaine de stage

La formation visait non seulement à familiariser les participants avec les différents aspects de l'informatique, mais également à nouer des contacts entre les enseignants et les hautes écoles. Chaque participant était invité à passer une semaine de stage dans une des hautes écoles partenaires. Il pouvait profiter de cette semaine pour jeter les bases de son futur travail de diplôme, tout en vivant au plus près le quotidien d'une haute école orientée vers la recherche.

Travail de Diplôme

Chaque participant devait remettre un travail de diplôme, considéré comme un travail de maîtrise. Il était relativement libre dans le choix du sujet, le contenu et le niveau du travail devaient cependant correspondre à la filière formation suivie.

OCI : mise en œuvre du curriculum

Elaborer un concept de formation et un curriculum est une chose, les mettre en œuvre en est une autre. Il s'agissait d'associer des contenus aux différents modules et les transmettre avec un maximum de clarté aux enseignants de gymnase. Chaque module était placé sous la responsabilité d'un ou de deux professeurs d'une haute école. Les responsables de module ont fait des propositions, défini les compétences que les participants devaient avoir acquis à la fin de la formation et élaboré le cours sur la base de ces données. A titre d'illustration, nous présentons ci-après quelques exemples de mise en œuvre tirés des formations données en Suisse alémanique.

Dans les pages qui suivent, Helmar Burkhart et Martin Guggisberg présentent les contenus sélectionnés pour la branche principale « Internet et multimédia » et l'option « Visualisation », ainsi que les exemples utilisés pour illustrer ces contenus et les constatations faites lors de la réalisation.

Dans la partie consacrée aux pratiques d'excellence, Martin Lehmann explique l'approche générique de la didactique de la discipline qui a déterminé le choix des thèmes didactiques en Suisse alémanique. En Suisse romande, la didactique de la discipline « informatique » ne figurait pas au programme.

Exemples de pratiques d'excellence

Dans ce chapitre, nous présentons quelques cours donnés en Suisse alémanique. Nous avons choisi trois thématiques, assorties de neuf exemples au total, pour illustrer l'enseignement de certaines notions fondamentales de l'informatique lors de la formation EFI-CH. Ces notions font par ailleurs partie du programme des études d'informatique traditionnelles au niveau « bachelor » et « master ». A la fin de chaque exemple, nous résumons les constatations les plus précieuses pour nous – et les participants.

Thématique 1 : Apprendre à programmer dans le monde du multimédia

La question du rôle de la programmation dans un cours d'informatique et du langage de programmation à enseigner occupe les didacticiens de l'informatique depuis fort longtemps. Les défenseurs de l'approche ludique (« c'est le seul moyen de motiver la génération Nintendo ») s'opposent aux adeptes de l'approche pragmatique (« enseignons Java, c'est utile »). Or, la formation en informatique doit tenir compte d'un consensus fondamental existant dans des milieux informatiques, à savoir que l'informatique est bien plus que l'apprentissage de la programmation et la compréhension des algorithmes et des structures de données, quel que soit le langage de programmation utilisé, et que l'acquisition d'approches méthodiques est essentielle pour résoudre des questions complexes.

D'autre part, toute forme d'automatisation du travail sur ordinateur donne souvent naissance à des actions typiques, bien connues des programmeurs : répétition d'opérations, distinction de cas, enregistrement de résultats intermédiaires, etc. Pour cette raison, le module « Programmation et algorithmes », l'une des thématiques clés du curriculum, était placé au début du cycle de formation.

Enseignant dans des disciplines différentes, les participants au cours possédaient des connaissances préalables très diverses. Afin de mettre tout le monde sur un pied d'égalité, nous avons choisi un premier outil de programmation qu'AUCUN des participants ne connaissait (exemple 1).

En outre, nous avons à cœur de soutenir leur envie d'apprendre, ce qui nous a amenés à intégrer de nombreux éléments multimédias aux cours consacrés à la programmation. En technique informatique, que l'on traite des séries de nombres ou des pixels au moyen d'une boucle, l'algorithme est toujours le même. En revanche, pour les participants, il y a une différence, et de taille, puisque dans le cas des pixels, le résultat est immédiatement perceptible. Un exercice du genre « Mets ton prof la tête en bas » peut être très motivant et facile à résoudre pour les débutants si l'outil de programmation est bien choisi (cf. exemple 2).

Enfin, l'enseignement de l'informatique a pour but de placer l'approche scientifique, souvent très abstraite, dans un contexte intéressant. Ainsi, la mise en récit (angl. storytelling) et l'approche ludique sont aujourd'hui des techniques reconnues en didactique de l'informatique. Il existe pour cela d'excellents outils, similaires à un jeu de construction de type Lego, comme le montre l'exemple 3.

Exemple 1 : Comprendre la récursivité, dessiner des plantes artificielles

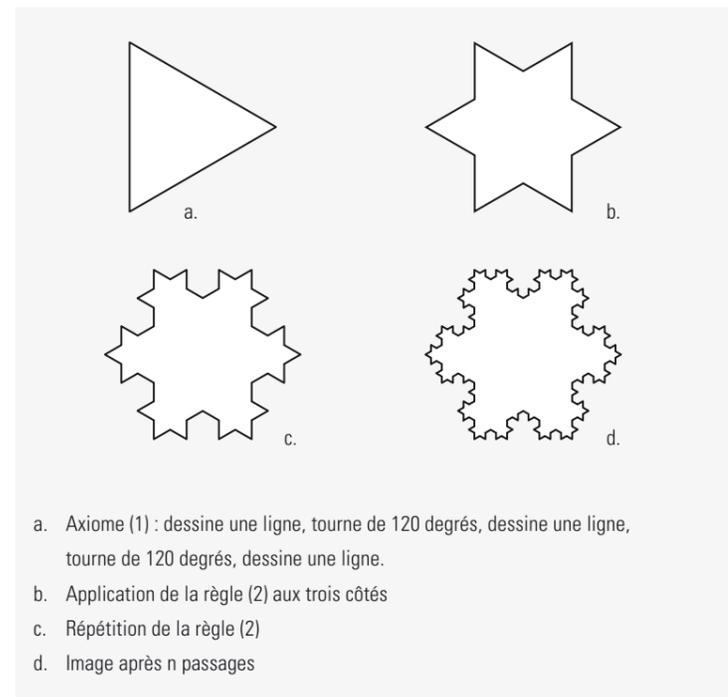
Le premier outil de programmation étudié dans le cadre de la formation EFI-CH n'était ni Java, ni Scratch, ni Alice, ni Kara, mais un langage appelé « L-System ». Inventé en 1968 par le biologiste hongrois Aristid Lindenmayer⁴, « L-System » permet de décrire le développement d'organismes biologiques.

Quelques règles simples suffisent pour produire des résultats surprenants. « L-System » est basé sur la notion de récursivité, autrement dit, l'application répétée d'un jeu de règles défini.

Les règles ci-dessous sont définies en premier et associées à une interprétation graphique :

F :	dessiner une ligne de la longueur définie dans la direction actuelle
+	tourner à gauche selon l'angle défini
-	tourner à droite selon l'angle défini

Si l'on pose comme point de départ (axiome) $F -- F -- F$ (1)
qu'on applique la règle de remplacement $F + F -- F + F$ (2)
et qu'on choisit un angle de 60 degrés,
le résultat est un *flocon de neige dit de Koch*. Au cours EFI-CH, l'applet Java PaLSystem⁵
a été utilisé pour piloter pas à pas les différents remplacements.



⁴ Aristid Lindenmayer: Mathematical models for cellular interaction in development. In: J. Theoret. Biology, 18, 1968, 280--315

⁵ www.javaview.de/vgp/tutor/lssystem/PaLSystem.html

Deux règles supplémentaires sont définies pour enregistrer la position et la direction :

[:	Enregistrer la position et la direction actuelles
]:	Rappeler la position et la direction actuelles à partir de la mémoire

Axiome	F	(3)
Règle de remplacement	F [+F] F [-F] [F]	(4)
et angle de	25.7 degrés	

Ces données permettent de créer, pas à pas, une plante artificielle simple.



Constatations

La programmation consiste à définir des règles formelles, exécutables par un automate.

Elle comprend deux niveaux : la manière de formuler (la syntaxe) et l'effet recherché (la sémantique). Ce dernier est le niveau le plus important.

Quelques règles concises permettent souvent de produire un grand effet.

Un ordinateur exécute systématiquement toutes les étapes demandées, sans se tromper, tandis que l'être humain commet facilement des erreurs lorsqu'il exécute la même tâche manuellement.

Exemple 2 : Le point sur les graphiques

La distinction entre graphiques vectorisés et non vectorisés, le codage d'informations de couleur, les principes de filtre et bien d'autres encore sont des éléments fondamentaux du multimédia que l'on retrouve fréquemment dans la pratique. Pour familiariser les participants avec ces notions, nous avons choisi lors de la formation EFI-CH un environnement de développement facile à utiliser, mais néanmoins performant : le logiciel open source Processing6, basé sur Java. Ce logiciel est déjà répandu, non seulement auprès des techniciens, mais également dans les milieux créatifs et artistiques.



Constatations

Les exercices multimédias sont des activités très motivantes et les participants acceptent plus facilement les exigences pointues liées à l'apprentissage d'un langage de programmation.

L'orientation événementielle, la création de graphiques et de vidéos peuvent être enseignées au niveau débutant.

La plateforme idéale pour les débutants ne révèle pas tout son potentiel d'emblée, mais permet d'accéder à l'ensemble des fonctions ultérieurement. Le passage de Processing à Java est possible sans rupture.

Le programme ci-dessous montre qu'il est possible de réaliser des effets graphiques avec des moyens simples. Le chargement d'une image, la définition de ses dimensions, l'inversion de l'ordre des pixels sont des opérations simples en soi, mais lorsqu'on les combine, elles permettent d'afficher une image à l'envers.

Exemple 3 : Raconte-moi une histoire

*Scratch*⁷ est un environnement de programmation graphique gratuit qui, par sa conception modulaire rappelant les briques Lego, permet d'aborder la programmation de manière simple et motivante. Il est possible d'obtenir rapidement un résultat encourageant par la simple combinaison de plusieurs éléments (briques). La prise en charge du multimédia (objets animés et sonores) et les commandes interactives via la souris et le clavier stimulent la créativité. Les programmes ainsi composés peuvent être publiés sur Internet, ce qui rend le résultat visible et réutilisable (ultérieurement).

En créant des programmes qui comprennent des actions exécutées simultanément sur plusieurs objets (« sprites »), le participant apprend et applique les notions liées au traitement des événements. Lors de la formation EFI-CH, les participants avaient pour mission de programmer la confrontation entre un chasseur et sa proie. L'énoncé très général de l'exercice laissait une grande place à la créativité, d'où des solutions très différentes : dans certaines réalisations, le chasseur attend que la proie pénètre dans son rayon d'action, puis il l'attaque, dans d'autres, la stratégie des chasseurs consiste à se diriger immédiatement vers la nouvelle proie.

La réalisation des simulations chasseur-proie représentait un véritable défi pour les participants. Il fallait analyser et comprendre chaque erreur de l'animal, puis trouver et évaluer de nouvelles solutions. Lors de la présentation de leurs solutions devant le groupe, les participants ont constaté qu'une stratégie qui pourtant avait fait ses preuves pouvait néanmoins échouer en fonction du contexte et du déroulement du jeu.

Constatations

Les éléments de programme visuels facilitent l'accès à la programmation.

Le traitement des commandes interactives reçues par le clavier, la souris et l'entrée audio est très exigeant en matière de modélisation et de programmation.

Souvent, plusieurs stratégies sont possibles pour un même jeu d'ordinateur.

Thématique 2 : Pas de pratique sans théorie

Les connaissances pratiques des outils informatiques sont uniquement utiles pour l'exécution de tâches de routine. En revanche, pour développer une fonction inédite, trouver sa propre modélisation à partir d'une idée et créer un produit sous forme de programme, il est indispensable de connaître les notions fondamentales de l'informatique. Même si l'étude et la compréhension des bases théoriques prennent du temps, ces dernières sont néanmoins indispensables pour concevoir un poste de travail informatisé efficient.

On se pose souvent la question de savoir quelle taille peut avoir un problème informatique afin qu'un ordinateur puisse calculer la solution dans un délai acceptable. L'exemple 4 donne quelques indications à ce sujet pour le domaine des algorithmes.

Dans certains cas, la première réalisation d'un modèle produit des résultats surprenants ou incorrects. L'exemple 5 montre que lors du calcul et de la modélisation scientifiques, il est important de tenir compte des propriétés fondamentales des architectures de calcul.

Afin de répondre aux attentes actuelles en matière d'interactivité et de comportement en temps réel, les programmeurs doivent maîtriser les bases de la programmation parallèle (exemple 6).

⁶ www.processing.org

⁷ <http://scratch.mit.edu>

Exemple 4 : Bien choisir son algorithme

Dans de nombreux cas, le « O-Kalkül[®] », l'un des sujets du premier bloc « Introduction à la programmation », permet de déterminer, avec une précision surprenante, le temps d'exécution d'un programme. L'algorithme utilisé est modélisé sous forme de fonction mathématique portant sur le nombre d'opérations effectuées en fonction de la taille du problème. En effet, en informatique, il existe toujours plusieurs méthodes différentes pour arriver à une solution, avec plus ou moins d'efficacité.

A titre d'exemple, examinons le problème suivant : soit une série de nombres entiers, positifs ou négatifs. L'exercice consiste à calculer la série de nombres consécutifs dont la somme est la plus élevée. Pour la suite de nombres 31 -41 59 26 -53 58 97 -93 -23 84, la somme maximale est 187, ce qui correspond à la somme de la série partielle comprise entre les nombres 59 et 97 (ces deux nombres inclus). Ce problème peut être calculé manuellement, puisque la série ne compte que 10 nombres. Mais un être humain ne parviendra pas à résoudre l'exercice si la série compte 1000 nombres ou plus.

Au cours EFI-CH, les participants ont vu différentes méthodes pour résoudre le problème de manière plus ou moins rapide. La solution la moins efficace et la plus efficace sont décrites ci-après :

- La méthode a) est une démarche minutieuse, qui calcule la somme de toutes les séries partielles et retient le montant maximal. Cette méthode est onéreuse, puisque le temps d'exécution augmente à la puissance 3. Si nous doublons le nombre d'éléments, le temps d'exécution est multiplié par huit.

- La méthode b) procède de manière ciblée. Le temps d'exécution augmente de manière linéaire : si la série compte deux fois plus de nombres, le temps d'exécution est deux fois plus long.

La réponse à la question suivante est donc facile à trouver : « Pour une série de 4000 nombres, le temps d'exécution est de 5,3 secondes avec la méthode a) et d'un centième de seconde en appliquant la méthode b). Quel est le temps de traitement pour chaque méthode si la série compte 8000 nombres ? » Les temps d'exécution calculés théoriquement, de 42 secondes et de 2 centièmes de seconde respectivement, correspondent de manière assez précise à ceux mesurés dans la pratique. Cette constatation a suscité l'étonnement des participants et les a encouragés à apprendre les bases théoriques de ces algorithmes.

Constatations

Dans de nombreux cas, certaines conditions cadres concrètes, telles que le langage de programmation utilisé, peuvent être ignorées lors de la résolution d'un problème. La méthode choisie, à savoir l'algorithme, est déterminante.

L'informatique s'interroge notamment sur les algorithmes qui permettent de résoudre un problème de manière efficace.

En appliquant certaines méthodes, on peut prédire le temps nécessaire à un ordinateur pour résoudre un problème.

[®] Nous utilisons ici l'expression employée dans la version allemande du rapport.

Exemple 5 : Ne pas toujours faire confiance à l'ordinateur

L'ordinateur est généralement considéré comme une machine à calculer très précise. Rappelons cependant que les ensembles de nombres utilisés en mathématiques n'existent que de manière approximative en informatique. Ainsi, les nombres entiers sont uniquement réalisés dans une plage de valeurs finie, de sorte que le résultat de l'addition de deux très grands nombres peut être faux. Il incombe au programmeur ou à l'environnement de programmation d'identifier ce problème de *dépassement de capacité*.

Autre défi pour les débutants en informatique : le problème de la représentation interne des nombres dans l'ordinateur. Alors que l'extrait de programme à gauche

```
a=1; b=0;
while (b≠5)
  b=b+a; écris(b);
écris(„Fin du calcul“);
```

1
2
3
4
5

Fin du calcul

incrémente la variable de 1 à chaque passage et s'arrête lorsque la valeur 5 est atteinte, le programme à droite (incrémenté par dixièmes) s'exécute de manière inattendue, car il ne s'arrête pas.

Cette différence de traitement est due au fait qu'il n'existe pas de représentation exacte de la valeur 0,1 dans l'ordinateur. Les nombres réels, tels que nous les connaissons en mathématiques, n'existent pas en informatique, où on parle de nombres machine. En raison de l'absence de représentation exacte, les tests d'égalité ou d'inégalité sont à proscrire, car le résultat est généralement faux.

```
a=0.1; b=0;
while (b≠5)
  b=b+a; écris(b);
écris(„Fin du calcul“);
```

...
4.800000000000000
4.900000000000000
5.000000000000000
5.100000000000000

...

Constatations

On peut apprendre une recette (par cœur), mais on en tire profit seulement si on comprend le contexte.

Les lois et les conditions des mathématiques ne sont pas toujours valables pour l'ordinateur.

Une petite modification dans un programme peut avoir des effets néfastes.

Exemple 6 : Le parallélisme, une discipline reine maîtrisable

Un être humain peut faire plusieurs choses à la fois : prendre des notes en téléphonant ou encore gesticuler en parlant. Grâce à la technologie AJAX, les applications Web modernes peuvent réagir très rapidement aux différentes interactions de l'utilisateur. Les requêtes sont envoyées sur Internet en temps réel et les résultats recueillis directement intégrés à la page Web actuelle. Cette interactivité exige la présence, en arrière-plan, de nombreux programmes qui réagissent à la survenue de l'événement qu'ils surveillent.

En programmation Web, le défi particulier réside dans le dynamisme des programmes. En effet, le déroulement effectif d'un programme n'est pas séquentiel, mais peut varier en fonction du comportement de l'utilisateur. Afin de familiariser les participants avec les principales notions de la programmation parallèle, de nombreux exercices sur le traitement des événements avaient déjà été présentés dans le bloc « Introduction à la programmation ». Pour illustrer la thématique, les participants ont étudié un problème standard, celui du « Dîner des philosophes ». Grand classique parmi les algorithmes, cet exercice montre comment un interblocage (« deadlock ») se produit dans le déroulement de processus.

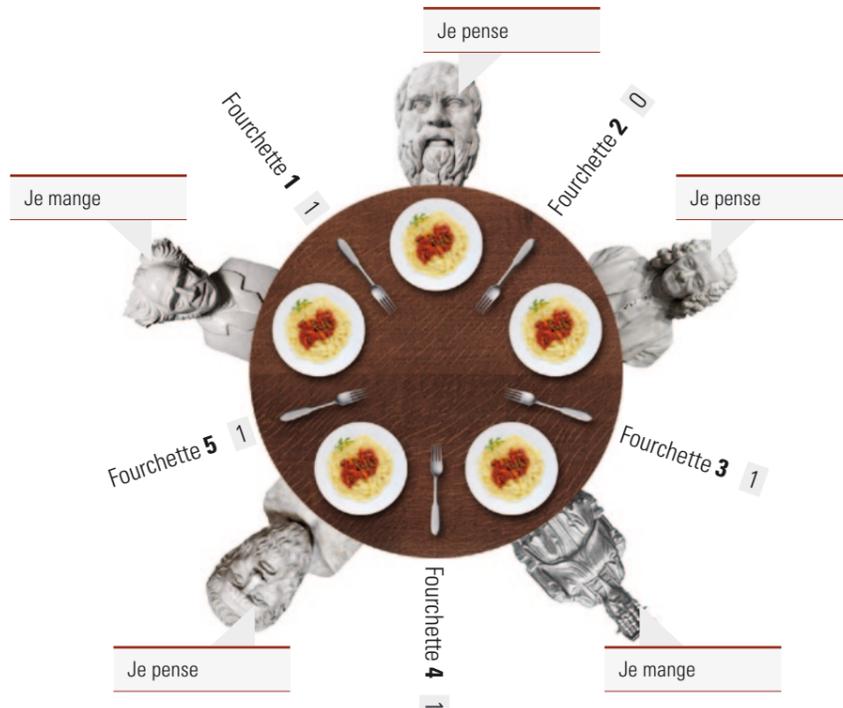
En résumé, le problème est le suivant : cinq philosophes sont réunis autour d'une table ronde. Chacun a devant lui une assiette de spaghettis. Pour manger les spaghettis, chaque convive a besoin de deux fourchettes en même temps. Or, il n'y a que cinq fourchettes, disposées à côté des cinq assiettes. Manifestement, des conflits de ressources peuvent se produire. Deux philosophes assis côte à côte ne peuvent donc pas manger en même temps. Toute solution allant dans ce sens serait fautive.

Constatations

Les systèmes informatiques interactifs modernes maîtrisent le traitement parallèle d'événements.

Le mode de pensée parallèle et les notions correspondantes peuvent être présentés à l'aide d'exemples parlants.

Afin d'utiliser le matériel informatique actuel et futur de manière plus efficiente, des connaissances de la programmation parallèle sont indispensables.



Thématique 3 : D'accord pour suivre les tendances en informatique, mais pas pour suivre la mode

De nombreux appareils mobiles sont aujourd'hui équipés de capteurs GPS à haute précision. Pour cette raison, le cours d'informatique doit aborder les notions et les possibilités de la géolocalisation, sans toutefois mettre en avant un fabricant ou des produits particuliers. L'exemple 7 présente des « geotracks » (itinéraires) individuels. L'objectif de cet exercice était de sensibiliser les participants, et à travers eux, leurs futurs élèves, à la gestion de leurs propres données et de leur fournir les connaissances techniques nécessaires à l'utilisation des nouvelles possibilités. Dans l'exemple 8, les participants ont intégré dans leur propre page Web des informations provenant d'une base de données publique. La visualisation en 3D fait appel, d'une part, à la compréhension de la technique sous-jacente et, d'autre part, à des ressources de calcul très importantes pour représenter les images de synthèse. L'exemple 9 montre différents processus de création d'animations en 3D.



GeoTracks⁹

Exemple 7 : Où suis-je ?

Les capteurs GPS des outils de navigation, des smartphones ou des caméras numériques mémorisent la position actuelle de l'utilisateur avec une précision de quelques mètres. Les informations géographiques enregistrées peuvent être réutilisées dans différentes applications, par exemple pour mesurer les performances sportives ou revoir ultérieurement le chemin parcouru lors d'une randonnée. Au cours d'une après-midi consacrée à la géolocalisation, les participants ont planifié leur propre balade, enregistré des images et des textes pour l'illustrer, puis publié le tout sur Internet.

Par ailleurs, on trouve sur Internet des randonnées ou des parcours VTT à télécharger sur un appareil de navigation ou un smartphone.^{10, 11}



Randonnée en 3D¹²

Constatations

La réalisation d'un projet de géolocalisation exige des connaissances du fonctionnement d'un capteur de localisation (généralement un GPS) et de la gestion des coordonnées sphériques.

Rechercher un endroit déterminé, suivre une balade culturelle ou mesurer les performances sportives et analyser les données recueillies sont des activités de géolocalisation en rapport direct avec la vie de tous les jours

⁹ http://www.everytrail.com/fullscreen.php?trip_id=532046

¹⁰ <http://www.gpswandern.de>

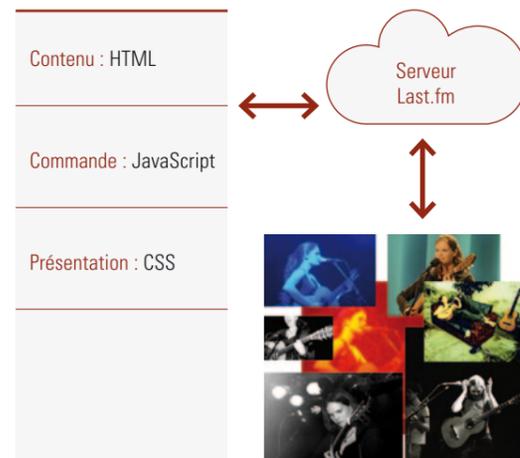
¹¹ <http://www.gps-tracks.com>

¹² <http://bit.ly/3dwandern>

Exemple 8 : Les données sont de l'argent, mais interconnectées, elles valent de l'or

L'application et l'utilisation des technologies Internet, ainsi que la création de contenus ont connu une évolution fulgurante au cours de ces dernières années. Le terme « mashup » (application composite) désigne la collecte d'informations provenant de sources de données différentes sur Internet et la combinaison de ces données sur une page Web. Dans le domaine des services cartographiques en particulier, les possibilités de visualisation de données en temps réel sont nombreuses, par exemple afficher les appartements vides à l'aide du service cartographique de Google¹³, suivre en direct le trafic aérien au-dessus de Zurich¹⁴ ou faire un tour virtuel en voiture en visualisant le paysage simultanément par la vitre et à vol d'oiseau¹⁵.

Les participants étaient invités, entre autres, à créer leur propre page Web interactive. L'un des projets avait pour but la réalisation d'une application qui, après la saisie du nom d'un interprète dans une base de données de musique accessible, puisse rechercher toutes les photos d'une taille donnée et les intégrer directement à la page Web créée par le participant¹⁶. Les pages Web interactives de ce type utilisent la technologie AJAX : après le déclenchement d'un événement (en règle générale par un clic sur un bouton), une requête est envoyée à un serveur sur Internet. Ce dernier envoie sa réponse dans un format normalisé (XML ou JSON). Le contenu de la réponse est alors intégré à la page Web personnelle et s'affiche immédiatement.



Les graphiques montrent les composants d'une page Web interactive. Les contenus de la page sont définis dans la partie HTML ; dans la plupart des cas, la page comprend des zones vides dans lesquelles s'afficheront les réponses recueillies.

La conception (mise en page, formatage des polices) est réalisée dans la partie CSS, tandis que les éléments de commande (logique) sont aujourd'hui programmés en JavaScript. Le nombre d'interactions possibles ne cessant d'augmenter, les développeurs de pages Web font souvent appel à un cadre d'applications (angl. « framework ») qui propose les fonctionnalités courantes souhaitées.



Constatations

Actuellement, une page Web ne peut plus se passer de fonctionnalités telles que l'interactivité et l'intégration automatique de données actuelles.

Pour programmer une page Web, il est indispensable d'avoir une vue d'ensemble et de comprendre le déroulement dans le temps des différents segments de programme.

La recherche et l'élimination d'erreurs sont un défi majeur de la programmation d'applications Web interactives.

¹³ www.housingmaps.com

¹⁴ <http://radar.zhaw.ch/radar.html>

¹⁵ <http://www.mapchannels.com/RouteMaps.aspx>

¹⁶ http://bit.ly/th_mueller

Exemple 9 : L'informatique 3D, pas seulement pour le cinéma

Le module à option « Visualisation » a permis aux participants de se familiariser avec les outils de visualisation open source PoV-Ray et Blender et de les utiliser en laboratoire pour réaliser leur propre projet d'environnement 3D. Certains s'étaient d'ailleurs tellement pris au jeu qu'ils avaient poursuivi le travail jusque tard dans la nuit. L'un de ces projets, intitulé « Domino Dome », est tout à fait exceptionnel. Il s'agit d'un arrangement de centaines de dominos et de plusieurs éléments mobiles. Dans l'animation très réaliste, un moteur physique intégré¹⁷ fait tomber les dominos. Le calcul de la simulation a duré plusieurs jours.

L'approche visuelle ludique a été une source d'inspiration pour l'un des participants qui a réalisé un travail de diplôme sur ce sujet. Il a créé une extension pour l'outil de visualisation Blender, qui permet de visualiser des biomolécules complexes (p. ex. des protéines) dans le langage visuel scientifique des biologistes et de représenter leurs mouvements au moyen d'une animation. La figure à droite présente les premiers résultats de ce travail.

Un autre travail montre qu'il est possible de créer des films spectroscopiques en 3D au moyen d'outils open source, dans une qualité très proche de celle des films 3D pour le cinéma. Ce projet¹⁷ a également inspiré un article scientifique pour la conférence INFOS 2011 (Informatik in der Bildung)¹⁹.



Constatations

Une communauté open source très active propose de nombreux médias pour une utilisation professionnelle (outils, modèles 3D et didacticiels).

L'informatique peut être un véritable outil de création esthétique-affective.

La création de visualisations 3D de haute qualité et leur comparaison avec les réalisations de professionnels de la science et des médias est une expérience très motivante.

¹⁷ http://www.youtube.com/watch?v=Lg_P6q0hX48

¹⁸ <http://goodpractice.epistemis.com/dna.html>

¹⁹ Beat Trachsler, Martin Guggisberg and Martin Lehmann. GI-Edition - Lecture Notes in Informatics (LNI), P-189, INFOS 2011, Bonner Köllen Verlag (2011) ISSN 1617-5468, ISBN 978-3-88579-283-3.

Exemples de pratiques d'excellence dans le domaine de la didactique de la discipline

La didactique de la discipline « informatique » a pour objet les méthodes d'enseignement de l'informatique. De ce vaste champ d'investigation, nous examinons ci-après deux questions essentielles.

Au cours d'informatique, tant la connaissance des notions théoriques que celle des produits jouent un rôle important. Si le cours est trop axé sur les produits, les connaissances transmises ne sont pas durables et, de plus, elles sont vite oubliées. Par contre, un cours exclusivement théorique manque de références à la mise en œuvre et les élèves auront du mal à appliquer dans la pratique les connaissances acquises.

Plus que les autres disciplines spécialisées, l'informatique est présente dans tous les domaines de notre vie. Elle couvre un champ très vaste, de la recherche aux applications. Comment rendre un cours d'informatique attrayant en tenant compte de toutes ces dimensions ?

Idées fondamentales

Vu l'évolution très rapide de l'informatique, le cours d'informatique risque de trop se focaliser sur les nouvelles technologies et d'oublier les contenus fondamentaux de la discipline. Le concept d'« idée fondamentale » forgé par Jerome Bruner²⁰ peut servir d'outil pour la sélection des contenus à enseigner. Les didacticiens Andreas Schwill²¹ et Werner Hartmann²² ont précisé le concept d'idée fondamentale en l'assortissant de cinq critères concrets :

Critère horizontal : l'idée est utilisable et identifiable sous différentes formes et dans différents domaines
Critère vertical : l'idée est comprise et peut être transmise à tous les niveaux intellectuels
Critère temporel : l'idée est perceptible à travers l'évolution historique du domaine et valable à long terme
Critère de sens : l'idée fait référence au mode de pensée et à la langue de tous les jours et à l'environnement de vie
Critère de représentation : l'idée peut être représentée à différents niveaux cognitifs (symbolique, iconique, énonciatif)

²⁰ Bruner, J.S. The process of education, Cambridge Mass. 1960
²¹ Schwill, A. Fundamentale Ideen der Informatik, Zentralbl. für Didaktik d. Mathematik (1993), p. 20-31.
²² Hartmann, W., Näf, M., Reichert, R. : Informatikunterricht planen und durchführen. Springer, 2006

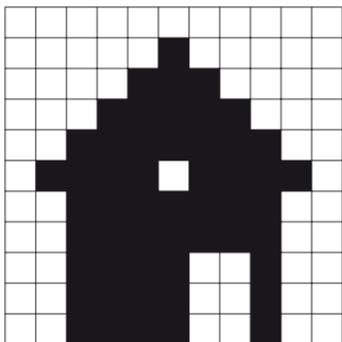
Exemple d'une compression sans perte : le codage des répétitions (RLE)

En rendant « visibles » les idées fondamentales, l'enseignant augmente les chances que son cours se concentre sur des notions durables et que ses élèves comprendront plus facilement des faits complexes.

La compression (avec ou sans perte) peut servir d'exemple d'idée fondamentale :

Nous utilisons cette technique dans différentes situations, par exemple en rédigeant un résumé, en faisant une valise ou en écoutant un morceau de musique en mp3 (critère horizontal). La compression peut être expliquée, de manière plus ou moins détaillée, à des personnes de tout âge (critère vertical). Depuis toujours, des procédés de compression intelligents permettent d'exploiter au mieux les ressources disponibles en quantité limitée (critère temporel). De même, la compression est aujourd'hui indispensable dans la plupart des canaux d'information (critère de sens). Les procédés de compression sont faciles à représenter sous forme textuelle ou visuelle. Lorsque nous faisons nos bagages, nous utilisons une forme très concrète de compression (si possible sans perte) (critère de représentation).

a) sans compression : 121 signes



W W W W W W W W W W W
W W W W W S W W W W W
W W W W S S S W W W W
W W W S S S S S W W W
W W S S S S S S S W W
W S S S S W S S S S W
W W S S S S S S S W W
W W S S S S S S S W W
W W S S S S W W S W W
W W S S S S W W S W W
W W S S S S W W S W W

b) compression par ligne : 72 signes

11 W
5 W 1 S 5 W
4 W 3 S 4 W
3 W 5 S 3 W
2 W 7 S 2 W
1 W 9 S 1 W
2 W 7 S 2 W
2 W 7 S 2 W
2 W 4 S 2 W 1 S 2 W
2 W 4 S 2 W 1 S 2 W
2 W 4 S 2 W 1 S 2 W

c) image en noir et blanc avec une compression aussi forte que possible : 29 signes. Pouvez-vous reconstituer l'image ?

16 1 9 3 7 5 5 7 3 4 1
4 3 7 4 7 4 4 2 1 4 4
2 1 4 4 2 1 2

L'enseignement basé sur des idées fondamentales permet la présentation de contenus durables, qui ont un sens pour les élèves et, en même temps, représentent un défi pour eux.

Great Principles of Computing

L'informatique est un domaine vaste et complexe, mais que l'on réduit souvent au seul aspect de la programmation. L'enseignement de l'informatique doit avoir pour but, entre autres, de permettre aux élèves de percevoir l'étendue et la complexité de l'informatique et d'aborder certains thèmes en profondeur. Denning²³ propose une structuration spécifique possible de l'informatique, qui repose sur sept « fenêtres » (« windows ») :

Computation: These principles address what processes, natural and artificial, are computational, what they can and cannot do, and how we cope with inherent and pervasive computational complexity.

Communication: These principles concern the transmission of data with reliable reception.

Coordination: These principles concern how autonomous entities work together toward a common result.

Recollection: These principles concern how computations store and recall information, and how data layout in the storage system affects their performance.

Automation: These principles concern finding efficient computational ways to perform human tasks. Tasks can be mental, such as doing arithmetic, playing chess, and planning schedules, or physical, such as running an assembly line, driving a car, controlling an airplane.

Evaluation: These principles concern how computing systems perform under various computational loads and how much capacity they need to deliver their results on time.

Design: These principles concern how to design software and computing systems that are dependable, reliable, usable, safe, and secure.

Ces sept « fenêtres » composent une structure qui, au cours d'informatique, peut être complétée au fur et à mesure par des thèmes et des exemples. Le site Web de SwissEduc²⁴ propose quelques exemples de matériel pédagogique basés sur les « Great Principles of Computing ».

Le fait de travailler selon une structure donnée permet de construire un ensemble de repères grâce auxquels les élèves pourront ordonner les nouveaux développements, sans que l'évolution fulgurante de l'informatique ne déforme leur vision.

²³ Denning, P. J. : Great Principles of Computing. Communications of the ACM, 46(11) : 15-20, Nov 2003.

²⁴ <http://swisseduc.ch/informatik/great-principles.html> (11.11.2011)

OCl : évaluation des cours en Suisse romande

En Suisse romande, les cours OCl ont été organisés à deux reprises seulement. Pour cette raison, une évaluation a été menée à la fin de chaque module. Les participants ont été invités à évaluer, pour chaque module, les formateurs, la structure, le matériel et l'utilisation de médias, à faire part de leurs observations positives et négatives, et à soumettre leurs propositions d'amélioration.



Evaluation du module de base

Le module de base correspondait dans une large mesure aux attentes de la grande majorité des participants, d'où des réactions très favorables. Le fait que la matière a été enseignée pratiquement « d'un seul tenant » par le professeur Petitpierre et M. A. Maurer est l'une des raisons de l'évaluation positive de ce module.

Evaluation des modules d'approfondissement

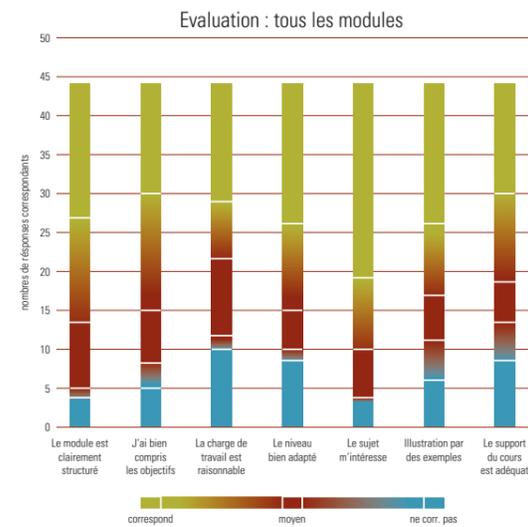
Il n'est pas question de taire ici l'ambiance très agitée dans laquelle s'est déroulée la première édition des modules DAS (modules d'approfondissement 1 à 5). En effet, certains participants au cours estimaient que le fait qu'ils étaient eux-mêmes enseignants n'était pas suffisamment pris en compte. De plus, ils souhaitaient des cours qui présentent un rapport plus direct avec la branche OCI qu'ils étaient censés enseigner par la suite. Les professeurs des hautes écoles, par contre, entendaient transmettre des connaissances allant au-delà de la matière enseignée ultérieurement. Manifestement, il y avait une grande divergence entre les attentes d'un certain nombre de participants et les objectifs initiaux de la formation.

Les vives discussions et les évaluations ont eu un effet positif : lors de la deuxième édition des modules d'approfondissement, la communication avait été améliorée et certaines suggestions avaient été mises en œuvre. Le résultat souhaité ne s'est pas fait attendre : l'évaluation de la deuxième édition était nettement plus favorable.

Les modules plus proches de la pratique (bases de données, programmation) ont généralement reçu un accueil plus favorable que les modules à fort caractère théorique (informatique théorique), à l'exception notable du module « Multimédia et visualisation ». Les connotations de cette désignation sont tellement fortes que, dans ce cas, l'écart entre les attentes et les contenus enseignés était particulièrement marqué. De plus, le module avait été confié à trois professeurs, dont l'un avait tenté, dans un premier temps, d'enseigner en anglais, ce qui avait également suscité une certaine irritation. De manière générale, les participants ont préféré les modules « d'un seul tenant ».

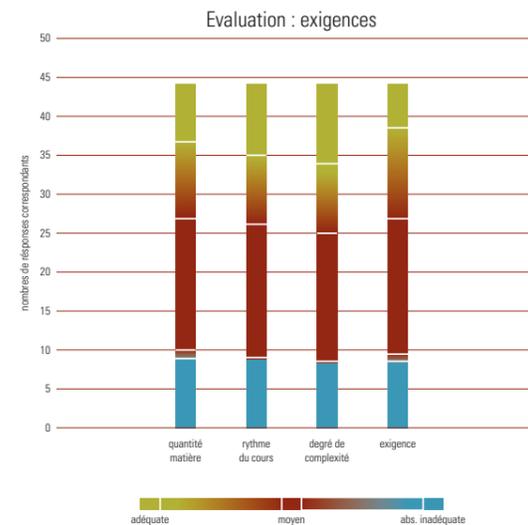
Malgré les discordances apparues au début de la formation, l'évaluation du cours était globalement positive.

Le graphique ci-dessous montre l'évaluation de la formation par les participants. Il cumule les commentaires relatifs aux deux éditions du cours et à tous les modules.



Les zones jaunes se rapportent aux évaluations positives, les rouges aux évaluations neutres. Seules les réponses en bleues étaient négatives. Si l'on tient compte de tous les modules et des deux éditions, on constate que les réponses positives sont majoritaires, malgré les problèmes mentionnés en introduction.

Les participants pouvaient également indiquer si les exigences à leur égard étaient adéquates ou non (du point de vue du participant). Le graphique ci-dessous montre que dans ce cas également, les attentes ont été remplies pour la plupart des participants.



Evaluation de la semaine de séminaire (cours bloc)

Les deux éditions du cours bloc ont été évaluées de manière globalement positive, malgré la complexité des thèmes abordés et le fait que certains participants avaient passé (de leur plein gré) une bonne partie de la nuit devant l'ordinateur. Les participants ont également apprécié d'avoir séjourné ensemble pendant une semaine dans un hôtel au bord du lac de Morat, un environnement qui a permis de travailler de manière plus intensive et de consolider le groupe.

En résumé

Enseigner à des enseignants est une entreprise à la fois difficile et gratifiante. Les enseignants sont motivés et ouverts à la nouveauté si le concept est compréhensible, si les attentes sont bien gérées et les structures claires. L'hétérogénéité des participants mentionnée plus haut se reflète également dans la perception très différente de la structure du curriculum et ce, jusqu'à la fin du cycle. Il est également intéressant de retenir, pour d'éventuelles manifestations ultérieures, que les modules « d'un seul tenant » – enseignés par un seul professeur – sont mieux accueillis que des modules confiés à plusieurs professeurs. Cette constatation a également été faite en Suisse alémanique.

Relevons également un autre enseignement positif : les participants ont communiqué leurs critiques aux professeurs des hautes écoles avec beaucoup d'assurance, ce qui a permis de tirer rapidement profit des difficultés rencontrées au cours de la première année et de procéder aux améliorations nécessaires.

De manière générale, on peut affirmer que la plupart des participants étaient satisfaits du cours, ce que bon nombre d'entre eux ont également confirmé lors d'entretiens personnels.

OCI : conclusion

- Deux facteurs ont contribué à la réalisation en un temps record du projet OCI : la forte motivation des personnes impliquées et le soutien financier de la Fondation Hasler. Les participants aux différents cycles de cours ont reçu une solide formation en informatique.

Or, il ne faut pas s'arrêter en si bon chemin. Pendant les différents cycles de cours, la SSIE (Société suisse de l'informatique dans l'enseignement) a lancé une initiative de mise en réseau, qu'elle intensifiera encore à l'avenir. Elle propose aux enseignants d'informatique des plateformes d'échange d'expériences et de matériel de cours ainsi que des formations continues.

Sans l'engagement d'un partenaire solide, qui a financé non seulement le développement et l'organisation de la formation, mais également la décharge des enseignants pour la durée des cours, le projet n'aurait pas pu être réalisé. La mise en place rapide d'une option complémentaire informatique devait répondre à un véritable besoin, même si la motivation de voir naître une nouvelle branche ou de suivre une formation continue était bien réelle. Il fallait donc trouver des participants au profil adéquat et les convaincre de soutenir le projet. En outre, la participation de professeurs engagés a été un facteur de succès déterminant. La formation n'aurait pas pu voir le jour sans leur motivation et leur conviction qu'à titre de professeurs de hautes écoles, ils pouvaient contribuer à instaurer l'informatique au gymnase. Au final, treize hautes écoles ont participé au projet EFI-CH, un regroupement de forces qui ne va pas de soi.

Professeurs et participants ont bénéficié de l'effet de synergie né de la collaboration entre les hautes écoles de Suisse alémanique, du Tessin et de Suisse romande. Ainsi, lors d'un stage dans une des hautes écoles, les participants à la formation ont pu se familiariser avec les différents domaines de recherche et les cultures d'enseignement de ces établissements.

Au vu des expériences généralement positives, certaines hautes écoles sont prêtes à proposer, à l'avenir également, des cours de formation continue pour les enseignants d'informatique, afin que les développements futurs de la science informatique soient également portés dans les écoles. La SSIE entend s'engager dans cette voie pour ses membres. Aussi souhaite-t-elle que l'informatique soit présentée comme une branche forte et intéressante dans l'offre de formation continue du Centre suisse de formation continue des professeurs de l'enseignement secondaire.

Perspectives

- Comme nous l'avons mentionné dans l'introduction de ce rapport, l'informatique a été intégrée aux plans d'études des gymnases suisses comme 14e option complémentaire lors de la petite révision du RRM95 en 2007.

La nouvelle branche a été accueillie à bras ouverts : en 2008/09, l'option complémentaire informatique (OCI) avait été proposée dans 27 écoles de 9 cantons différents. Au final, elle a été mise en place dans 26 écoles. Forts d'une longue tradition informatique, quatre cantons alémaniques faisaient partie des pionniers de l'OCI. Les cantons d'Argovie, de Berne, de Thurgovie et du Valais pouvaient compter sur des enseignants qui avaient fait des études d'informatique (comme branche principale ou secondaire) auparavant et qui pouvaient désormais enseigner leur propre discipline. En Suisse romande, les cantons du Jura et du Valais ont été les premiers à lancer l'OCI en 2008. En quatre ans, le nombre d'écoles instaurant l'informatique en option complémentaire n'a cessé d'augmenter. Le projet EFI-CH, lancé à un moment propice, a contribué de manière décisive au succès de l'OC Informatique. Grâce aux enseignants supplémentaires ayant suivi une formation en informatique, d'autres cantons ont également pu proposer la nouvelle option complémentaire à leurs gymnasiens.

Le tableau ci-dessous montre que pour l'année scolaire en cours²⁵ (2011/12), l'OCI figure au programme de 114 des quelque 150 gymnases. Dans la quatrième colonne du tableau, la lettre a) se réfère aux cantons qui ont décidé l'introduction générale de l'OC Informatique pour l'année scolaire figurant en première colonne ; la lettre b) aux cantons qui ont créé, à titre de projet pilote cantonal, une OC Informatique dans certains gymnases.

Tout connaisseur du système fédéraliste suisse sait que cette récapitulation doit faire l'objet de certaines réserves : d'une année à l'autre, la mise en place d'une option dépend du nombre d'inscriptions ; dans certaines écoles, l'effectif minimal est de 4 élèves, d'autres exigent 8 élèves. La taille des classes est tout aussi variable : certains cantons acceptent de scinder une classe dès 11 participants, dans d'autres, une classe d'option complémentaire peut compter jusqu'à 24 élèves.

Toutefois, les statistiques des écoles proposant l'informatique comme option complémentaire permettent également de constater que l'OCI est choisie par 10 à 15 % des élèves, avec un effectif féminin moyen par classe d'environ 18 %. Ce taux montre que, d'une part, la nouvelle option répond à l'une des revendications implicites du rapport EVAMAR I, à savoir que l'offre gymnasiale devrait s'enrichir de branches attrayantes, afin d'inciter plus d'élèves masculins à suivre des études gymnasiales²⁶. D'autre part, la faible participation féminine indique également que les écoles ont la possibilité de susciter l'intérêt pour la technique auprès des jeunes filles en proposant une branche scientifique supplémentaire.

Année scolaire	Nombre d'écoles Premier cours OCI	Nombre total d'écoles avec une OCI	a) Introduction dans le canton : b) Introduction dans les différents établissements :
2008/09	26	26	a) AG, BE, JU, TG, VS b) LU, SO, GR, ZH
2009/10	37	63	a) BL, BS, FR, NE, OW, SG b) GE, ZG, ZH
2010/11	41	104	a) AR, GE, NW, SZ, TI, VD
2011/12	10	114	b) GL, UR, ZH

Tableau 7 : Nombre de gymnases proposant l'option complémentaire Informatique

En rétrospective, l'introduction de l'informatique comme option complémentaire au gymnase est une réussite, la 14e branche semble bien acceptée. Ce succès est dû à la réalisation du projet EFI-CH, et, dans une large mesure, aux nombreux enseignants motivés qui ont contribué à la conception d'une nouvelle discipline scolaire. Il s'agit maintenant de nourrir cette motivation. Plusieurs mesures ont déjà été engagées à cet effet : le projet de mise en réseau lancé en 2008 offre une plateforme d'échange et de développement aux enseignants d'informatique, qui souvent sont seuls dans leur établissement à enseigner cette matière. De plus, les professeurs et d'autres formateurs impliqués dans le projet EFI-CH sont prêts à s'engager à l'avenir également dans le domaine de la formation continue en informatique et à proposer une offre de perfectionnement à long terme en collaboration avec la SSIE. Ils contribuent ainsi à assurer la qualité des contenus de l'OC Informatique.

Leur offre montre que l'engagement des partenaires de la coopération EFI-CH, jusqu'ici unique, s'inscrit dans une optique de durabilité et se poursuivra à long terme. Au cours des prochaines années, les différentes écoles continueront à s'investir pour une informatique forte dans les gymnases suisses.

²⁵ Informations sur le nombre de gymnases en Suisse et détails supplémentaires sur les options complémentaires ainsi que l'OC Informatique et son introduction en particulier, cf. Peter, J. (2009). L'informatique comme option complémentaire. Rapport sur l'état de l'introduction de l'informatique comme option complémentaire et la mise en réseau des enseignant-e-s de l'informatique. www.oc-informatique.ch (11.11.2012).

²⁶ EVAMAR I, L'essentiel en bref 2004, p. 6, L'essentiel en bref : http://www.sbf.admin.ch/evamar/berichte/EVAMAR-L_essentiel_en_bref_fr.pdf

Prestataires et partenaires

— Prestataires EFI-Deutschschweiz

Université de Zurich – Institut für Informatik (Lead)
 Université de Bâle – Departement für Informatik
 Université de Fribourg – Departement für Informatik
 Hochschule Luzern – Forschung und Entwicklung in Informatik
 Pädagogische Hochschule Zentralschweiz Luzern – Zentrum Medienbildung
 Pädagogische Hochschule Bern – Institut Sekundarstufe II.

Prestataires OCI-Suisse romande

Ecole Polytechnique Fédérale de Lausanne (EPFL) – School for Computer and Communication Science (Lead)
 Université de Fribourg – Département d'informatique
 Université de Genève – Département d'informatique
 Université de Neuchâtel – Institut d'informatique.

Prestataires OCI-Ticino

Università della Svizzera italiana (USI) – Facoltà di scienze informatiche (Lead)
 Scuola universitaria professionale della Svizzera italiana (SUPSI) – Dipartimento tecnologie innovative
 Eidgenössische Technische Hochschule Zürich (ETH) – Departement Informatik

Partenaire

Centre suisse de formation continue des professeurs de l'enseignement secondaire (WBZ CPS)

Financement

Fondation Hasler

_oci

Option complémentaire informatique

Prof. André Schiper
EPFL IC IIF LSR
INF 236 (Bâtiment INF)
Station 14
CH-1015 Lausanne