

DIDACTIQUE COMPAREE ET ENSEIGNEMENT DE L'INFORMATIQUE

Fabienne Viallet, Patrice Venturini

Université de Toulouse
DiDiST CREFI-T, Didactique des Disciplines et Techniques
Bâtiment 3R1, B2
118, Route de Narbonne
31 077 Toulouse Cedex 14
fabienne.viallet@iut-tlse3.fr
patrice.venturini@bv.toulouse.iufm.fr

Mots-clés : Didactique de l'informatique, didactique comparée, enseignement et apprentissage de la boucle.

Résumé : La didactique comparée reposant sur le concept de pluralité, nous proposons son ouverture vers une discipline nouvelle, l'informatique. Nous présentons donc une étude de l'enseignement d'un objet de savoir paradigmatique de la discipline informatique, la boucle. Cette étude débute par le programme épistémologique qui nous permet d'identifier le savoir de référence, se poursuit par l'analyse transpositive du texte du savoir et du savoir enseigné pour aboutir à la mise à disposition des éléments culturels et institutionnels nécessaires à la réalisation d'une analyse de l'action conjointe de l'enseignement apprentissage de cet objet de savoir. L'informatique étant une discipline jeune, dans laquelle aucune étude théorique centrée sur la notion de savoir n'a encore été menée, cette étude nous a conduits à interroger son épistémologie.

1. Problématique

L'informatique ne s'étant jamais constituée en France comme discipline d'enseignement scolaire¹, elle n'a réellement bénéficié des avancées de la recherche en didactique qu'entre 1988 et 1996 (Baron, 2009)². Un état de l'art que nous avons mené sur les travaux existants en didactique de l'informatique, tant du côté français qu'anglo-saxon, nous a montré que les recherches menées étaient essentiellement centrées sur l'étude des difficultés rencontrées par les étudiants dans un contexte donné afin de proposer des solutions adéquates aux enseignants. Suivant le problème évoqué, le cadre théorique, les outils et méthodes de recherche changent. Ainsi, dans leur ouvrage de synthèse sur la didactique de l'informatique, Sally Fincher et Marian Petre (2004) montrent qu'il existe un problème dans la définition même du contour de cette discipline qui évolue sans cesse, emprunte à d'autres disciplines, propose une très grande variété d'enseignements et que le domaine ne possède donc pas encore de théorie propre permettant le développement de travaux de recherches unifiés. Quelle que soit la nature des travaux, le savoir en jeu dans l'enseignement,

¹ En 1970, sous la présidence de Georges Pompidou, il a été décidé de ne pas créer de nouvelle discipline informatique (avec un CAPES, une agrégation, un corps d'inspection générale, des programmes inspirés de la recherche et de l'enseignement supérieur), mais de former à l'informatique des enseignants issus de différentes disciplines afin de permettre à l'informatique de pénétrer l'ensemble des disciplines existantes. (Pair, 1987). Aujourd'hui encore, l'enseignement obligatoire de l'algorithme en seconde générale est dévolu aux enseignants de mathématiques (Eduscol, 2009).

² Depuis 2003, elle s'est orientée vers les problèmes liés à l'apprentissage de logiciels, au travers notamment des colloques Didapro.

cœur de la discipline, est toujours questionné mais peu analysé. D'une façon générale, l'absence de travaux théoriques proposant une entrée par le savoir ne permet pas de le caractériser.

Pourtant, l'évolution de l'informatique dans la société est telle que son enseignement est aujourd'hui présent dans toutes les formations de l'enseignement supérieur et du secondaire. Une étude centrée sur les savoirs supportés par cette discipline semblant donc être intéressante à mener, nous proposons de construire un cadre conceptuel pour mener cette étude, en se situant dans une approche comparatiste et en empruntant aux outils que la didactique a déjà expérimentés dans d'autres disciplines.

2. Construction d'un cadre conceptuel d'étude

L'identification de ce que recouvre l'informatique en tant que discipline reste une question toujours vive, qui se déporte sur son enseignement. En 1985, Janine Rogalski suggérait comme piste de recherche possible, l'étude du savoir informatique au travers de la transposition didactique. Optant pour une dichotomie entre savoirs savants et pratiques sociales de référence, elle pensait cette question en termes de dualité entre la transposition d'une organisation conceptuelle de la discipline informatique et la transposition des pratiques professionnelles de conception et de « suivi » des programmes et systèmes informatiques. Cette question duale entre organisation conceptuelle et pratiques professionnelles a été débattue dans la littérature de l'époque. Deux réponses complémentaires ont été apportées à cette question :

- En 1994, Yves Chevallard étend la notion de savoir à ses pratiques de référence et à son domaine de réalité. Samuel Johsua (1997 et 1998) a modélisé cette métonymie en proposant une organisation dynamique des savoirs dont la nature n'est figée ni dans le temps, ni dans les pratiques.
- Yves Chevallard (1994) déporte le débat sur la localisation des savoirs à enseigner, en transformant la question initiale de la nature du savoir à enseigner à celle de ses institutions de référence : à qui se réfère le savoir à enseigner ? Cette référence est-elle légitime ? Est-elle pertinente dans le cadre de la formation ? L'évolution de la nature des savoirs est alors envisagée comme des processus de transformations et de négociations entre trois institutions : l'institution utilisatrice, l'école et une institution savante.

Une étude transpositive consiste donc à identifier les multiples références auxquelles se réfère l'objet de savoir enseigné, quelque soit leurs natures (savoirs savants ou pratiques sociales de référence), tout en repérant les dynamiques en jeu entre les différentes institutions qui ont contribué à la construction de son enseignement. Une telle étude pose ainsi la question de la légitimité et de la pertinence de l'objet de savoir enseigné. Elle situe le débat au cœur de l'anthropologie des savoirs, également nommée par Yves Chevallard (1991), l'épistémologie du savoir.

La question d'une double étude transpositive du savoir informatique basée sur la différenciation de la nature des savoirs, perd donc sa pertinence. En regard des différents travaux de recherche portants sur la transposition didactique, l'étude transpositive du savoir informatique revient donc à identifier les savoirs et les processus mis en œuvre lors de la construction d'un enseignement en informatique et à déterminer ses différents habitats en termes d'écologie des savoirs, dans une perspective épistémologique. Ainsi, cette étude applique au savoir informatique les questions propres à l'anthropologie des savoirs : de quels savoirs de référence est issu l'enseignement de l'informatique ? Pourquoi est-il enseigné ? Ces questions non encore abordées en didactique de l'informatique, sont pourtant au cœur des débats portants sur « la nature épistémologique de l'informatique en tant que discipline d'enseignement supérieure » (Bruillard, 2009).

D'un point de vue didactique, l'étude transpositive telle que proposée par Yves Chevallard a montré au fil du temps à la fois sa puissance et ses limites. Ainsi, dans une étude détaillée de travaux sur la transposition didactique dans plusieurs disciplines, Maria Luisa Schubauer-Leoni et Francia Leutenegger (2005) ont montré que cette approche transpositive n'était pas suffisante pour

rendre compte de l'ensemble des processus transpositifs. Elles proposent donc d'articuler à ce programme épistémologique descendant, un programme didactique ascendant partant de la situation de classe, en utilisant le cadre théorique de l'action conjointe (Sensevy, 2007). Ces deux programmes restant complémentaires au programme cognitif destiné, lui, à repérer les difficultés liées à l'apprentissage. Cependant, le programme didactique ne peut pas être conduit sans que soit mené le programme épistémologique permettant d'identifier au cours de l'analyse, les éléments de références des objets de savoirs : « le mouvement ascendant du programme didactique à visée comparatiste pourrait souffrir, quand à lui, d'une perte de vue de l'organisation des objets de savoir et se trouver ainsi démuné pour saisir l'espace culturel et institutionnel dont disposent effectivement les sujets pour agir » (Maria Luisa Schubauer-Leoni et Franca Leutenegger 2005, page 421).

Ainsi, en didactique comparée, réaliser une étude transpositive sur le savoir informatique, consiste à mener un programme épistémologique à articuler avec un programme didactique. Etant donné qu'il n'existe pas de travaux théoriques sur l'enseignement de l'informatique proposant une entrée par le savoir et permettant de le caractériser, il nous semble opportun de débiter par le programme épistémologique afin de pouvoir construire une analyse a priori d'une activité informatique possible et attendue dans une institution donnée.

3. Notre étude : méthodologie et premiers résultats

3.1 Définition de l'objet d'étude

Pour mener cette étude, nous avons donc identifié un objet de savoir qui est à la fois représentatif de notre domaine d'étude (l'informatique), enseigné de façon courante et suffisamment complexe pour pouvoir donner lieu à une analyse. Dans la communauté informatique, l'algorithmique et la programmation sont considérées comme étant au cœur de la discipline³. Certains de leurs concepts traversent tous les paradigmes de programmations et sont considérés comme fondamentaux. C'est le cas notamment des structures de contrôle. Elles permettent, au travers d'énoncés proposés par le langage de programmation, de contrôler le flot de données dans les programmes (Lecarme, 2007). La littérature spécialisée montre que leur acquisition constitue un des obstacles pour l'apprentissage de la programmation.

Parmi les différentes structures de contrôle existantes, nous avons choisi d'étudier l'itération qui permet de faire exécuter au programme une même séquence d'instructions, un certain nombre de fois. Tous les langages de programmation proposent des énoncés spécifiques, appelés « boucle », qui permettent d'implémenter l'itération sans faire appel à la récursivité. La boucle nous semble être la structure de contrôle la plus intéressante à étudier. En effet c'est un des premiers objets de savoir construit en informatique, qui est commun à tous les langages de programmation et enseigné de façon systématique dans les formations de base en informatique dans le monde. Bien que l'informatique soit une discipline en perpétuelle évolution, la boucle reste un des concepts qui perdure et traverse les années tout en restant central à la discipline. C'est également un des objets d'enseignement sur lequel des travaux de recherche conséquents ont déjà été menés (Ginat 2004, Méjias 1985, Samurçay 1985, Rogalski 1985). La boucle constitue donc pour nous un bon candidat pour une étude transpositive.

3.2 Analyse épistémologique du savoir

L'absence de travaux en épistémologie de l'informatique nous amène à démarrer préalablement à notre étude transpositive, une analyse épistémologique de la boucle afin de la caractériser et

³ Cf. le curriculum des ACM (<http://www.acm.org/education/curricula/ComputerScience2008.pdf>, consulté le 27 Août 2009) et le PPN (<http://media.education.gouv.fr/file/77/6/776.pdf>, consulté le 18 Juillet 2010)

d'identifier le savoir de référence sur lequel nous pourrions baser notre étude transpositive. En partant des fondements de la transposition didactique, nous avons abordé cette étude par l'étude du savoir savant. Ainsi, en parcourant la littérature informatique, nous avons recherché les articles publiés disponibles dont l'objet principal est la boucle. Parmi tous ces articles, nous avons effectué un tri pour isoler les articles les plus pertinents. Nous avons utilisé deux critères pour réaliser cette sélection : l'ancienneté et le nombre de références. Nous avons également utilisé des ouvrages conçus a posteriori à partir d'une compilation d'articles jugés fondamentaux tels que « Structured Programming » (Dijkstra, 1979).

Cette recherche bibliographique nous a permis d'identifier, de caractériser et de localiser le savoir de référence de la boucle. Nous avons ainsi découvert que la question de l'écriture de la boucle, qui possède plusieurs syntaxes plus ou moins structurantes pour le programme, a suscité de vifs débats dans la communauté dans les années 65-70 et a soulevé la question fondamentale de la validité des résultats produits⁴. A l'issue de ce débat, la boucle a été plébiscitée comme étant la structure de contrôle à privilégier pour réaliser des itérations, aux dépens des branchements inconditionnels qui ne permettent pas l'écriture lisible et vérifiable des programmes. Jacopini et Böhm (1966) ont montré par ailleurs que la séquence, la sélection et la boucle étaient les 3 structures de contrôle nécessaires et suffisantes pour écrire tous les programmes. Ce débat a ainsi contribué à l'avènement de la programmation structurée et des preuves de programme.

Le résultat de cette analyse épistémologique basée sur l'écologie des savoirs, montre que les savoirs savants générés à l'issue du débat sur l'écriture de la boucle, sont institués par des organismes de recherche (institutions savantes) eu égard aux problèmes rencontrés par les informaticiens industriels et chercheurs, dans leur pratique de la programmation (institution utilisatrice). La question qui surgit alors concerne l'institution scolaire : que fait-elle de ce nouveau savoir ? Comment le transpose-t-elle ?

3.3 Analyse transpositive du savoir enseigné

Pour réaliser l'analyse transpositive, nous avons sélectionné un lieu d'investigation institutionnel (une école) où la boucle est enseignée dans le cadre d'un cours suivant un curriculum national : il s'agit du département informatique de l'IUT de Toulouse où la boucle est enseignée en première année universitaire dans le cadre d'un cours d'algorithmique dont le contenu est défini par le Programme Pédagogique National⁵ (PPN). Cet enseignement se compose d'un cours magistral, de travaux dirigés et de travaux pratiques réalisés sur ordinateurs.

A partir du PPN, le curriculum de cette formation, nous avons réalisé une analyse de type transposition didactique externe sur l'enseignement de l'itération. Ainsi nous avons mis en tension ce texte du savoir avec le savoir de référence que nous avons identifié lors de la phase précédente. Nous avons alors observé le phénomène de dépersonnalisation du savoir – les trois structures de contrôle de base (Böhm, 1966) sont par exemple citées sans être reliées à leur source – et une forte désyncrétisation associée à la programmabilité des savoirs. Les savoirs à enseigner sont répartis dans des Unités de Formation correspondant chacune à un volume horaire et à un ensemble de prérequis. Et les problèmes de validité des programmes informatiques (ayant pour origine la boucle), sont abordés sous trois angles – la qualité des programmes, les outils de trace et de mise au point et la validité des programmes – mélangés dans trois unités de formation non reliées. L'analyse du contenu du PPN nous montre ainsi qu'il n'existe aucun lien entre la genèse socio-historique du savoir de référence (et notamment des débats générés dans la communauté) avec l'enseignement de la boucle.

⁴ Dans sa leçon inaugurale au collège de France, Gérard Berry, le premier informaticien à intégrer cette institution, explique qu'une des spécificités de l'informatique est « le bug » (Berry, 2008).

⁵ <http://media.education.gouv.fr/file/77/6/776.pdf>, consulté le 18 Juillet 2010.

Notre analyse du PPN nous a également permis d'identifier la noosphère (composée des chefs de département, d'enseignants et d'industriels) qui élabore le texte du savoir à enseigner en appliquant les directives ministérielles. Les trois institutions (savante, utilisatrice et école) sont donc représentées. Elles garantissent la légitimité des savoirs et inscrivent ce texte au sein de la société française contemporaine. La construction des niveaux de codétermination didactique (Chevallard, 2004) présentée figure 1, nous permet d'identifier et de hiérarchiser les contraintes sociologiques d'élaboration de ce texte et nous donnent des indications sur certains choix effectués au cours de la transposition didactique. Dans une perspective comparatiste, ces niveaux permettent la construction de critères de comparaisons.

La suite de notre travail a consisté à réaliser l'analyse transpositive du savoir enseigné en cours magistral, en utilisant le support de cours, qui représente ce que l'étudiant doit avoir (au mieux) écrit à l'issue de son cours magistral. L'absence d'interaction entre l'enseignant et les étudiants dans ce type d'enseignement, nous a conduits à ne pas utiliser la théorie de l'action conjointe (Sensevy, 2007) et à nous centrer sur l'analyse du texte.

Nous avons donc mis en tension ce savoir à enseigner avec le savoir de référence et le texte du savoir issu du PPN, en nous attachant à repérer une fois de plus, les processus de dépersonnalisation, de désyncrétisation et de programmabilité. Nous constatons que ce savoir à enseigner étant très proche du texte du savoir, il reste donc éloigné du savoir de référence.

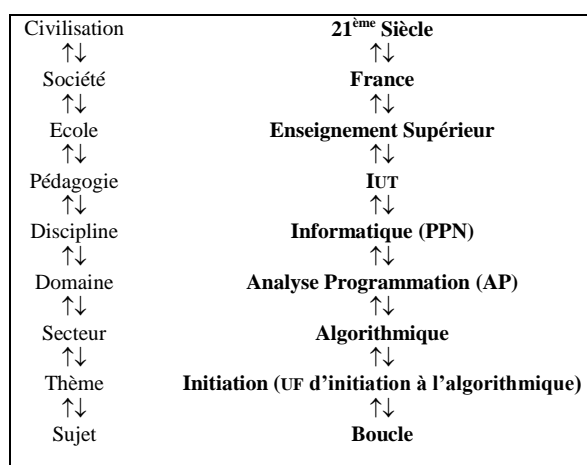


Figure 1 : Niveaux de codétermination didactique en informatique

Pour pouvoir rendre compte de ce savoir à enseigner, nous avons emprunté aux concepts de nature des savoirs en jeux (notion, notions proto informatique, ou notion para informatique), de chronogenèse de l'enseignement associée au temps didactique et d'identification des prérequis pour analyser le contenu. La figure 2 montre comment ce cours présente successivement les notions d'algorithme, d'acteur et de structures de contrôle en se basant sur la notion para informatique (uniquement préconstruites par monstration) d'action. De plus, ces notions sont présentées en faisant appel de façon implicite aux notions proto informatiques (compétences requises par les élèves dans le cadre du contrat didactique) d'ordinateur, de langage informatique et de raisonnement informatique.

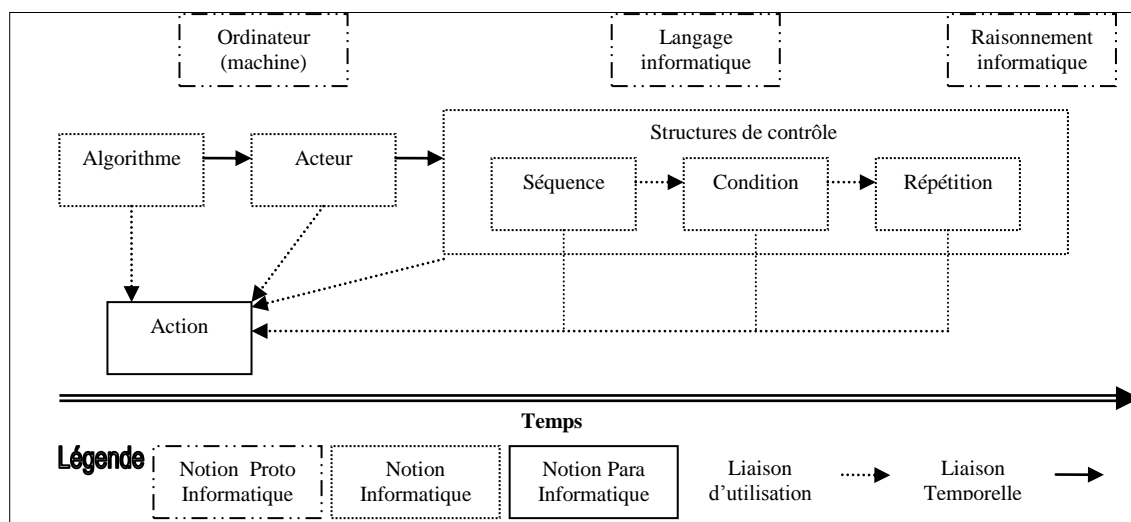


Figure 2 : Synopsis du savoir à enseigner

3.4 Le programme didactique

La poursuite de ce travail consiste à mener le programme didactique, c'est-à-dire à conduire une observation de la poursuite de cet enseignement en séance de TD où les interactions entre enseignants et étudiants permettent une co-construction des savoirs et donc une analyse de l'action conjointe. Cette analyse basée sur le système de description théorique de la TACD (Sensevy, 2008) pourra être conduite puisque les éléments culturels et institutionnels qui président à l'organisation de cet objet de savoir qui est la boucle, ont été identifiés.

4. Conclusion

La didactique comparée repose sur le concept de pluralité. Son ouverture vers une discipline nouvelle, l'informatique, lui offre la possibilité d'investiguer un nouveau champ de pratiques dont l'importance ne cesse de croître. Cependant, l'inscription de l'informatique au sein de la didactique comparée se heurte aux problèmes de la jeunesse de la discipline : l'identification de ce que recouvre l'informatique en tant que discipline reste en effet une question toujours vive et l'étude de son enseignement n'a pas encore donné lieu à des études théoriques centrées sur la notion de savoir. Ainsi, pour conduire notre projet visant à mener conjointement un programme épistémologique et un programme didactique sur un objet de savoir représentatif de la discipline que nous avons défini, la boucle, nous avons débuté par une analyse épistémologique de ce savoir. Nous avons ainsi identifié le savoir de référence et nous avons pu poursuivre par une analyse transpositive du texte du savoir et du savoir enseigné pour aboutir à la mise à disposition des éléments culturels et institutionnels nécessaires à la réalisation d'une analyse de l'action conjointe de l'enseignement apprentissage de cet objet de savoir. Nos travaux futurs devraient se poursuivre dans cette direction, en permettant l'analyse des séances de travaux dirigés sur cet objet de savoir. Ainsi cette étude nous a permis d'envisager l'étude d'enseignements d'informatiques sous l'angle des savoirs spécifiques à la discipline. Elle nous a notamment permis de construire un cadre d'analyse générique de l'enseignement de la boucle, qui peut être réutilisé pour réaliser l'analyse transpositive d'autres enseignements dans d'autres contextes, comme par exemple celui de l'algorithmique qui est délivré depuis l'an dernier au lycée en classe de seconde générale dans le cours de mathématique, par des enseignants non informaticiens. Enfin, ce travail nous a incidemment conduit à interroger l'épistémologie de la discipline, qui reste encore à construire.

5. Références et bibliographie

- Baron, G.L., Bruillard, E., Pochon, L.O. (2009). Enjeux didactiques de l'informatique et des ses outils vingt ans après : regards sur un cheminement. In G.L. Baron, E. Bruillard, L.O. Pochon (Ed), *Informatique et progiciels en éducation et en formation*. Lyon : INRP.
- Berry, G. (2008). *Pourquoi et comment le monde devient numérique*. Collège de France : Fayard
- Böhm, C. & Jacopini, G. (1966). Flow diagrams, turing machines and languages with only two formation rules. *Communications of the ACM*, 9, 5, 366-371.
- Brousseau, G. (1998). *Théorie des situations didactiques*. Grenoble : La pensée sauvage
- Bruillard, E. (2009). Place de l'informatique dans l'enseignement secondaire : réflexions introductives. In G.L. Baron, E. Bruillard, L.O. Pochon (Ed), *Informatique et progiciels en éducation et en formation*. Lyon : INRP.
- Chevallard, Y. (1985). Pourquoi la transposition didactique ? In Y. Chevallard (Ed), *La transposition didactique, du savoir savant au savoir enseigné*. Grenoble : La pensée sauvage.
- Chevallard, Y. (1991). *La transposition didactique, du savoir savant au savoir enseigné*. Grenoble : La pensée sauvage.
- Chevallard, Y. (1994). Les processus de transposition didactique et leur théorisation. In G. Arzac, Y., Chevallard, J.L., Martinand J.L., Tiberghin, A. (Eds), *La transposition didactique à l'épreuve*. Grenoble : La pensée sauvage. (pp. 135-180).
- Chevallard, Y. (2004). La place des mathématiques vivantes dans l'éducation secondaire : transposition didactique des mathématiques et nouvelle épistémologie scolaire. *3ème Université d'été Animath*, Saint-Flour (Cantal), 22-27 août 2004.
- Dijkstra, E. (1979). Structured programming. In E. N. Yourdon (Ed.), *Classics in Software Engineering*. (pp. 84-88). Upper Saddle River: Yourdon Press.
- EduSCOL. (2009). *Mathématiques. Lycées. Ressources pour la classe de seconde. Algorithmique*. http://media.eduscol.education.fr/file/Programmes/17/8/Doc_ress_algo_v25_109178.pdf, consulté le 19 Juillet 2010.
- Fincher, S. & Petre, M. (2004). *Computer Science Education Research*. London: Taylor & Francis Group plc.
- Ginat, D. (2004). On novice loop boundaries and Range Conception. *Computer Science Education*, 14(3), 165-181.
- Johsua, S. (1997). Le concept de transposition didactique peut-il étendre sa portée au-delà de la didactique des sciences et des mathématiques ? *Skholè*, 6.
- Johsua, S. (1998). *Des savoirs et de leur étude : vers un cadre de réflexion pour l'approche didactique. L'année de la recherche en sciences de l'éducation*. Paris : PUF.
- Lecarme, O. (2007). *Langages et paradigmes*. <http://deptinfo.unice.fr/~ol/M11/chap4-handout.pdf>, consulté le 17 Juillet 2010.
- Meijas, B. (1985). *Difficultés conceptuelles dans l'écriture d'algorithmes itératifs chez des élèves de collège*. Thèse de doctorat. Grenoble : université scientifique et médicale de Grenoble.
- Pair, C. (1987). Informatique et Enseignement : hier, aujourd'hui et demain. *Bulletin de l'EPI*, 47.
- Rogalski J. (1985). Alphabétisation informatique. *Bulletin APMEP*, 347, 61-74.
- Samurçay, R. (1985). Signification et fonctionnement du concept de variable informatique chez des élèves débutants. *Educational Studies in Mathematic*, 16(2), 143-161.
- Schubauer-Leoni M.L. & Leutenegger, F. (2005). *Une relecture des phénomènes transpositifs à la lumière de la didactique comparée*. *RSSE*, 27(3), 407-427.
- Schubauer-Leoni M.L. (2008). Didactique. In A. Van Zanten (Ed.), *Dictionnaire de l'éducation* (pp.129-133). Paris: PUF/Quadrige.
- Sensevy, G. & Mercier, A. (2007). *Agir ensemble. L'action conjointe du professeur et des élèves dans le système didactique*. Rennes : PUR.

Sensevy, G. (2008). Didactique comparée et générale. In A. Van Zanten (Ed.), *Dictionnaire de l'éducation* (pp.133-136). Paris: PUF/Quadrige.