Numéros / n° 7-8 - Culture du code

Â«Â La programmation comme outil créatif Â Â»

Sébastien Clara

Résumé

La programmation informatique nous permet de mettre en ?uvre nos idées musicales dans un contexte temps réel, mais aussi en temps différé. Après plus d?une décennie d?usage, nous constatons que la programmation n?est plus seulement un outil de production, mais est devenue aussi un outil de conceptualisation. Dans cet article, nous décrivons comment s?est réalisé ce glissement d?usage. Pour ce faire, nous aborderons la fonction informatique, la programmation orientée objet et nous présenterons comment nous utilisons la programmation comme outil créatif.

Introduction

Depuis nos premières expérimentations de glissement sémantique, chères à notre premier professeur de composition, Michel Meynaud, nous avons appris que ce type de glissement est courant, notamment dans la musique.

Durant la seconde moitié du xx^{e} siècle, par exemple, les musiciens du mouvement de la musique répétitive nord-américaine ont détourné des opérations compositionnelles issues des techniques de studio pour développer leurs musiques instrumentales. Terry Riley a utilisé le concept de la ligne à retard pour créer des accumulations, Steve Reich a adopté la boucle et le déphasage pour créer un discours rythmique évolutif et Philip Glass a employé le collage par concaténation pour développer des motifs mélodiques simples (Mertens, 1988).

L?école spectrale française a fait de même. Grâce aux avancées technologiques de leur époque, les musiciens de cette école ont pu disséquer intimement les sons et utiliser les résultats de leurs analyses en tant que modèle. Par exemple, Gérard Grisey a désigné ses techniques compositionnelles de synthèse instrumentale.

> *L?aspect technique de cette écriture qui synthétise des spectres complexes, articule leurs transitoires, joue sur les glissements insensibles d?une forme à une autre, souligne les sons résultants, prend les battements comme source rythmique, les filtrages et les déphasages comme source mélodique, pour ne citer que quelques traits parmi les plus saillants (Grisey, 1991, p. 353).*

Dans la suite de cet article, nous décrirons l?influence de la programmation sur notre méthode de création musicale. Pour ce faire, nous nous intéresserons d?abord à la notion de fonction en informatique, puis à la programmation orientée objet.

1. La fonction

En programmation, un sous-programme est une séquence d?instructions dédiée à l?exécution d?une tâche particulière. Elle est empaquetée dans une unité élémentaire qu?on qualifie de fonction, procédure, sous-routine ou méthode. Cette unité est appelée ensuite, à chaque fois que cette tâche particulière doit

être réalisée. Les programmeurs peuvent définir leurs sous-programmes dans leur programme, ou bien ils peuvent les consigner dans des bibliothèques dédiées, accessibles aux autres programmes.

Cependant, la structuration modulaire d?un code source n?est pas une obligation lorsque l?on élabore une application et on peut rédiger un programme sans avoir à le fragmenter en parties plus simples. Toutefois, il est généralement plus avantageux d?opter pour une programmation modulaire, dont certains sous-programmes seront développés spécialement à cet effet et d?autres seront tirés des bibliothèques de sous-programmes standards.

En effet, la décomposition d?un programme en sous-routines permet d?articuler ce dernier en différentes fonctions et de simplifier la programmation. Les fonctions sont alors autonomes et on peut les tester indépendamment du reste du programme. L?emploi d?une bibliothèque publique permet également de déléguer la réalisation de tâches standards et de se focaliser sur un problème particulier. Cette répartition allège considérablement les programmes et limite par conséquent les erreurs de programmation.
David John Wheeler a recommandé cette méthode de programmation en 1952 (Wheeler, 1952), mais comment a-t-il pu proposer ces optimisations d?écriture alors que les langages de programmation étaient en train d?apparaître ?

Dans la deuxième partie de son *Discours de la méthode*, publié en 1637, René Descartes procède à une introspection sur son système de raisonnement, afin de s?assurer que ses connaissances sont justifiées par des démonstrations strictes. Il souhaite ainsi s?affranchir de toute notion acquise par mimétisme ou habitudes coutumières. Pour entreprendre cette démarche, il se fixe quatre préceptes de conduite. Le but du premier précepte est de tester la robustesse de son raisonnement en remettant en causes ses certitudes. Le deuxième précepte consiste à dissoudre une difficulté en fragments plus simples, afin de faciliter la résolution du problème. Le troisième précepte, qui est le corollaire du deuxième, est de conduire un raisonnement par ordre croissant de difficulté. Enfin, le dernier permet de circonscrire les faiblesses d?un raisonnement en établissant un socle exhaustif d?arguments par des listes.

Dans son article sur l?usage des sous-programmes en programmation informatique, David John Wheeler préconise une structuration des programmes basée sur les préceptes prescrits par René Descartes. Pour justifier cette affirmation, reprenons par ordre d?énonciation ces derniers.

En effet, David John Wheeler propose une méthode pour améliorer l?efficacité de la programmation. Cette démarche d?optimisation démontre son introspection face à la structuration d?un programme et sa volonté de dépasser ses acquis. Cependant, cette démarche n?a été possible que par la remise en cause de ses certitudes, comme le préconise le premier précepte de Descartes.

L?usage de la fonction en programmation répond au deuxième et au troisième préceptes. Le rôle d?une fonction est de résoudre un problème élémentaire. Pour la mettre en ?uvre, il est donc préalablement nécessaire de désarticuler un problème complexe en sous-problèmes. De même, l?usage de la fonction pour développer une application repose sur une résolution par ordre croissant de difficulté. L?assemblage de plusieurs fonctions permet d?en réaliser de plus complexes et ce processus de construction pyramidale aboutit à la création d?une application.

De nombreuses tâches standards sont répertoriées dans des bibliothèques publiques afin de simplifier la programmation et d?accélérer les développements. Il existe par exemple des bibliothèques dédiées à la création d?interface graphique. Ces dernières listent de nombreux éléments graphiques et les méthodes pour interagir avec eux. Suivant le problème que nous rencontrons, nous pouvons utiliser une bibliothèque telle quelle et, le cas échéant, nous pouvons rédiger des fonctions particulières pour répondre à des spécificités. Selon nous, la gestion en bibliothèque des fonctions repose sur le quatrième précepte de Descartes. L?exhaustivité et l?optimisation des fonctions d?une bibliothèque impactent directement la rapidité du développement, la qualité, voire le nombre des fonctionnalités de l?application qui l?utilise.

Nous déduisons de ce procédé de raisonnement une première méthode créative de type vertical. La première étape consiste à énumérer des opérations élémentaires. En combinant les éléments de ce premier ensemble, nous construisons un deuxième ensemble d?opérations plus complexes, et ainsi de suite. À la fin de ce processus, la structure généalogique d?une opération complexe repose sur une arborescence d?opérations qui est conduite par ordre croissant de complexité.

Néanmoins, le *Discours de la méthode* est d?obédience positiviste : il est en effet sous-titré *Pour bien*

conduire sa raison, et chercher la vérité dans les sciences. En opposition à ce courant de pensée, Jean-Louis Le Moigne critique la méthode proposée par Descartes, dans son ouvrage sur *La théorie du système général*, en se référant aux épistémologies constructivistes et notamment aux travaux menés par Jean Piaget, Herbert Simon et Edgar Morin (Le Moigne, 1977). Pour ce faire, il oppose la perspective ensembliste (ou réductionniste) de Descartes à la célèbre maxime d?Aristote : « La totalité est plus que la somme des parties » et conclut son premier chapitre en proposant une approche analytique systémique suivant quatre préceptes (1).

Toutefois, le premier précepte de Le Moigne suggère de contextualiser son propre raisonnement. De fait, nous ne pouvons pas nous passer de l?approche ensembliste qui structure la pensée des ingénieurs, comme nous venons de le constater avec l?article de David John Wheel. Dans ce cas, comment traiter les composants d?une problématique par ordre croissant de difficulté, sans cloisonner notre raisonnement dans une structure hiérarchique arborescente ? En d?autres termes, comment libérer notre créativité ?

Une solution de haut niveau à cette question consiste à fixer une limite entre la réalisation d?un objet et son usage. Le tout d?Aristote serait pour nous l?application ou l?objet final de cette construction ensembliste et l?usage final de ce tout n?est pas monolithique. Dans son article sur *Les détournements de jeux vidéo par les joueurs* (Barnabé, 2015), Fanny Barnabé expose différents usages de réappropriation d?un jeu vidéo. La construction ensembliste d?un jeu vidéo aboutit à des usages protéiformes, plus variés que ce à quoi l?on pourrait s?attendre (2). De même, en musique, Brian Eno dans *Under stars* (1983), Tristan Murail dans *Atlantys* (1984) ou le groupe norvégien A-ha dans *Take On Me* (1984) utilisent tous le synthétiseur numérique emblématique DX7 commercialisé par Yamaha pour interpréter leur pièce, alors qu?ils ont développé des esthétiques différentes.

En informatique, la fonction permet de traiter des données, mais comment organiser ces dernières ? La programmation orientée objet apporte une réponse organisationnelle, mais aussi une solution de plus bas niveau au dilemme auquel nous sommes confrontés entre la pensée de Descartes et celle de Le Moigne.

2. La programmation orientée objet

La programmation orientée objet apparaît à la fin des années soixante, bien que des chercheurs affirment que des techniques orientées objets rudimentaires sont mises en ?uvre dans la conception d?un missile dès 1957 (Ten Dyke et Kunz, 1989). Toutefois, l?essor de ce paradigme de programmation s?opère au début des années quatre-vingt, avec le développement actif du langage orienté objet Smalltalk.

> *L?orientation objet permet de modéliser le système sous forme d?objets interagissant les uns avec les autres. Ainsi, quel que soit le type de système que l?on modélise, on le considérera comme un ensemble d?objets reliés entre eux d?une manière ou d?une autre. [?] Ce que les objets modélisent dépend donc de ce que nous voulons représenter avec notre modèle objet (Jacobson, 1993, p. 45).*

Ce paradigme de programmation propose une approche « naturaliste », puisqu?un programme orienté objet reflète la nature du problème traité. Grâce à l?objet, qui impose une syntaxe particulière, la programmation et la maintenance d?un code source sont facilitées. La syntaxe d?un langage orienté objet permet d?*encapsuler* les données et les fonctions dans un objet. De plus, elle est *polymorphique*. En effet, un traitement comparable sur des données de natures dissemblables est nommé par la même expression. Par exemple, les opérations de transposition d?une série de notes MIDI et d?un échantillon sonore ne relèvent pas du même mécanisme algorithmique, alors que la méthode pour effectuer l?opération sur ces deux éléments porte la même dénomination. En outre, les caractéristiques d?un objet peuvent être transmises à un autre par un procédé d?*héritage*. Cette possibilité permet d?accroître rapidement le potentiel d?un langage, tout en gardant une compatibilité avec la syntaxe précédente.

Les spécificités (encapsulation, polymorphisme, héritage) d?un langage orienté objet structurent l?organisation d?un programme. Cependant, plus le concept objet a gagné en maturité, plus l?intérêt des informaticiens s?est déplacé vers l?analyse objet. Sous le terme de programmation orientée objet (et en règle générale pour toute activité de programmation) se retrouvent trois disciplines distinctes : l?analyse, la conception et la programmation.

L?analyse est la première étape et elle permet de prendre connaissance de la problématique à traiter. Elle

délimite un champ d?investigation et met en évidence le problème. La conception résout le problème par la construction d?une solution informatique et la mise en place d?une modélisation adaptée. La programmation est la dernière étape et elle consiste seulement à traduire la précédente étape dans un langage de programmation qui suit la spécificité de l?environnement d?implémentation.

Par conséquent, les artistes qui utilisent la programmation pour développer leurs ?uvres devraient suivre ces préconisations méthodologiques et segmenter leur travail créatif en trois étapes distinctes (3). Toutefois, bien qu?indispensables en informatique, ces étapes ne sont pas applicables *stricto sensu* en musique, du fait de la nature de cette activité. En effet, l?étape d?ajustement rétroactive du programme est incompatible avec la méthodologie de l?informatique, mais nous pensons qu?elle est indispensable en musique, afin de valider par l?écoute la pertinence musicale d?une formalisation quelconque. En cela, nous adoptons une démarche analogue à celle d?Aristoxène de Tarente, en effectuant un raisonnement déductif à partir des données saisies par notre système auditif. Dans la section suivante, nous détaillons notre méthode.

3. La création orientée objet

À partir des années quatre-vingt, Horacio Vaggione a développé une méthode compositionnelle orientée objet (ou par réseau d?objets). En effet, il considère l?usage de l?informatique dans son activité compositionnelle comme une « pluralité d?opérations diverses, plutôt qu?un seul algorithme » (Vaggione, 2003, p. 97). Le paradigme objet de l?informatique lui permet d?articuler son outillage compositionnel hétérogène et l?ouvre à « l?idée de multiplicité morphologique » (Budón, 2007, p. 103), afin de faire proliférer son matériau compositionnel de base, qui peut être par exemple un algorithme, une partition ou un échantillon sonore, en utilisant les opérations conceptuelles d?encapsulation, d?héritage, de polymorphisme sur des strates temporelles différentes.

La construction ensembliste verticale de la fonction informatique trouve une échappatoire horizontale avec le paradigme objet. La création des objets est réalisée en encapsulant des fonctions et des données, mais l?usage protéiforme des objets par héritage et polymorphisme ouvre un univers opératoire qui n?est plus cloisonné par une structure hiérarchique de type arborescent. Pour ce faire, l?énumération de tous types d?opérations élémentaires et de données et leur interconnexion dans un système décentralisé de type réseau avec des liaisons qui ne sont pas préétablies permettent d?accroître notre créativité et de penser ce qui nous était impensable.

Pour notre propre travail compositionnel, nous utilisons exclusivement comme outil le langage de programmation orienté objet SuperCollider (4). Cet outil est dédié au traitement, à la synthèse sonore temps réel et à la composition algorithmique. En utilisant ce langage, nous employons implicitement la méthode d?Horacio Vaggione (5), qu?il a développée explicitement par une analogie méthodologique au paradigme de programmation orientée objet. En effet, nous ne sommes pas obligés d?utiliser un langage de programmation orienté objet pour conceptualiser et déployer le paradigme objet. Inversement, nous ne sommes pas obligés de réaliser un programme orienté objet lorsque nous utilisons un langage orienté objet.

Mettons en ?uvre cette méthode créative orientée objet afin de donner un exemple tangible. Pour ce faire, nous allons employer par héritage des opérations compositionnelles traditionnelles basées sur l?imitation. Toutefois, nous ne déploierons pas ces opérations sur des notes, mais sur un enregistrement sonore suivant une démarche polymorphe. Le but de cet exemple est de démultiplier notre matériau de base.

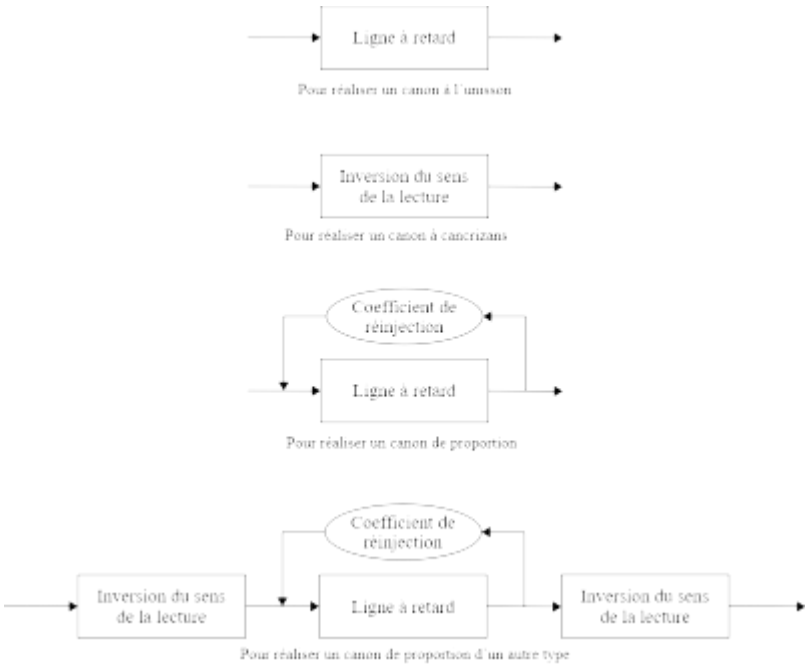
Pour produire un canon à l?unisson, nous appliquons à notre séquence originale un traitement composé d?une ligne à retard avec un délai suffisamment long pour produire un écho. Pour réaliser un canon à cancrizans, nous inversons le sens de lecture de la séquence originale et nous superposons cette nouvelle séquence à la première. Pour réaliser un canon de proportion, nous avons différentes méthodes (6) et cette diversité augmente les possibilités de prolifération de notre matériau de base. Pour cet exemple, cependant, nous restreignons la création de nos opérations aux traitements d?inversion du sens de lecture et à la ligne à retard que nous venons d?utiliser pour créer nos deux premières opérations.

En réduisant la durée du délai de la ligne à retard et en contrôlant par un coefficient une réinjection de la sortie de la ligne à retard à son entrée, nous obtenons un effet de réverbération (Roads, 1998, p. 110). Ce

traitement est généralement employé pour créer et appliquer un espace homogène sur des sources sonores d?origines hétérogènes. Dans notre cas, nous l?utilisons pour augmenter la durée de notre séquence originale (7) et produire un canon de proportion avec la fusion de cette dernière et de celle traitée par réverbération. En revanche, en utilisant une réverbération inversée, on obtient un canon de proportion d?une autre nature. L?opération de réverbération inversée consiste à inverser le sens de lecture de la séquence sonore, à la réverbérer et à l?inverser de nouveau. Récupérons la structure de cette dernière opération et remplaçons le traitement de réverbération par une ligne à retard dont la durée de déphasage est réglée pour produire un écho?

À partir de deux traitements élémentaires (l?inversion du sens de lecture et la ligne à retard, voir figure 1), nous mettons en ?uvre un héritage compositionnel sur un matériau fixé par l?enregistrement et nous obtenons cinq opérations. En ajoutant à notre réseau d?opérations la transposition, en modifiant les paramètres de nos traitements et en parallélisant les opérations, nous pouvons générer un matériau complexe à partir d?une idée qui, elle, ne l?est pas obligatoirement. De plus, pour cet exemple, nous nous sommes focalisés sur la génération de matériau, mais nous aurions pu resserrer l?usage de ces opérations pour générer des patrons polyrythmiques, réaliser des articulations de différentes formes, etc (8).

Figure 1. Traitements pour réaliser des opérations basées sur l?imitation



Source : Sébastien Clara, 2019

Conclusion

Selon Hugues Dufourt, le travail d?écriture compositionnelle consiste à refouler sa première impulsion musicale et à fonder son discours sur la cohérence des actes, sur des structures logiques, sur des conduites opératoires (Dufourt, 2004, p. 72). Dans le cadre d?un environnement compositionnel orienté objet, comme SuperCollider, il convient de traduire préalablement ses conduites opératoires dans des objets pour structurer son programme et de le faire également pour tout élément susceptible d?être utile au développement, comme les traitements, les synthétiseurs, etc. Puis, d?articuler un réseau « d?automates finis, où la communication se fait d?un voisin à un voisin quelconque, où les tiges ou canaux ne préexistent pas, où les individus sont tous interchangeables, se définissent seulement par un état à tel moment, de telle façon que les opérations locales se coordonnent et que le résultat final global se synchronise » (Deleuze et Guattari, 1980, p. 26).

Pour tisser de tels systèmes rhizomatiques, Gilles Deleuze et Félix Guattari préconisent six principes qui

supposent l?abandon des systèmes de type arborescent, dont la structure binaire de leur processus d?engendrement découle de la « pensée la plus classique et la plus réfléchie, la plus vieille, la plus fatiguée » (Deleuze et Guattari, 1980, p. 11).

Les principes de connexion et d?hétérogénéité permettent de nous décloisonner de nos *a priori*, afin d?effectuer des mises en relations invraisemblables et surprenantes. Le principe de multiplicité a pour but de s?affranchir de l?unicité et de l?universel par des processus de prolifération et d?hybridation. Le principe de rupture asignifiante permet de se libérer des limites, des bordures, des zones ou encore des frontières. « Un rhizome peut être rompu, brisé en un endroit quelconque, il reprend suivant telle ou telle de ses lignes et suivant d?autres lignes » (Deleuze et Guattari, 1980, p. 16).

Les principes de cartographie et de décalcomanie s?opposent, à première vue. En effet, le processus de cartographie est une activité en constante évolution. Il ne reproduit pas mais construit, et il nous aide à effectuer des connexions inattendues par les entrées multiples qu?il matérialise dans l?espace, tandis que le processus de décalcomanie n?est que mimétisme et cette unique activité n?est « que ruine de l?âme » (Rabelais, 1996, p. 123). Toutefois,

on sera souvent forcé de tourner dans des impasses, de passer par des pouvoirs signifiants et des affections subjectives, de prendre appui sur des formations ?dipiennes, paranoïaques ou pires encore, comme sur des territorialités durcies qui rendent possibles d?autres opérations transformationnelles (Deleuze et Guattari, 1980, p. 23).

Par conséquent, la fonction informatique, la méthode de Descartes ou l?imitation ne sont pas dépréciées, si nous les articulons dans une pensée rhizomatique. Dans notre cas, nous la mettons en ?uvre avec, comme support, la programmation orientée objet dans un cadre créatif.

---

1. « *Le précepte de pertinence* : Convenir que tout objet que nous considérerons se définit par rapport aux intentions implicites ou explicites du modélisateur. [...] *Le précepte du globalisme* : Considérer toujours l?objet à connaître par notre intelligence comme une partie immergée et active au sein d?un plus grand tout. [...] *Le précepte téléologique* : Interpréter l?objet non pas en lui-même, mais par son comportement, sans chercher à expliquer *a priori* ce comportement par quelque loi impliquée dans une éventuelle structure. [...] *Le précepte de l?agrégativité* : Convenir que toute représentation est partisane, non pas par oubli du modélisateur, mais délibérément. » (Le Moigne, 1977, p. 43).

2. La première forme de détournement étudiée par Fanny Barnabé dans son article est la fanfiction. Cette activité se définit comme l?écriture de récits inspirés d?univers fictionnels préexistants par les amateurs de ces derniers. La deuxième forme est la production de films d?animation humoristiques ou parodiques, de clips musicaux, de spectacles d?improvisation ou encore de longs métrages fictionnels réalisés à partir des moteurs graphiques des jeux. Ce type de production audiovisuelle est appelée machinima. La troisième forme de détournement est le modding. Cette activité consiste à modifier certains éléments d?un jeu dans le but d?en créer une version améliorée ou même, parfois, de produire un nouveau jeu à part entière. La dernière forme de détournement étudiée dans son article est le speedrun. Il s?agit de terminer un jeu le plus rapidement possible et de réaliser une vidéo de cette performance (Barnabé, 2015).

3. La popularisation de l?informatique est advenue durant les années quatre-vingt avec la commercialisation de l?ordinateur personnel. Des outils spécifiques à la musique ont été développés à cette période, comme la station de travail audionumérique, mais des langages de programmation ont aussi été conçus spécialement pour la création musicale. L?ambition de Miller Puckette par exemple, le concepteur de Max et de Pure Data, est de proposer un environnement graphique de programmation à l?attention des artistes (Puckette, 2011).

4. La première version de SuperCollider date de 1996 et elle a été développée par James McCartney. Depuis 2002, SuperCollider est publié sous licence GPLv2 et ce langage rassemble une large communauté de chercheurs, d?artistes et de musiciens.

5. Nous avons étudié et utilisé explicitement la méthode compositionnelle orientée objet d?Horacio Vaggione lors de la réalisation de notre mémoire de Master 2 à l?université Paris 8 que nous avons validé en 2014.

6. Notamment, en modifiant la vitesse de lecture de l?échantillon ou en utilisant les algorithmes d?étirement temporel (*time stretch*).

7. Pour contourner les limites technologiques de son époque et composer *Studie II* en 1953, par exemple, Karlheinz Stockhausen utilise la réverbération pour allonger la durée de fragments sonores. Pour ce faire, il enregistre sur bande magnétique des signaux sinusoïdaux monophoniques et crée une séquence en concaténant ces fragments de bande. Puis, il lit cette séquence avec un magnétophone, mais il accélère la vitesse de lecture et la traite par un effet de réverbération. Avec ce procédé, il ne produit pas un matériau mélodique, mais des structures sonores, des timbres particuliers. (Tissot, 2008, p. 168).

8. Pour des exemples concrets de programmation, référez-vous à nos publications (Clara, 2017, 2018 et 2019).

Pour citer ce document:
Sébastien Clara, Â« La programmation comme outil créatif Â», *RFIM* [En ligne], Numéros, n° 7-8 - Culture du code, Mis Ã jour le 21/12/2020
URL: http://revues.mshparisnord.org/rfim/index.php?id=580