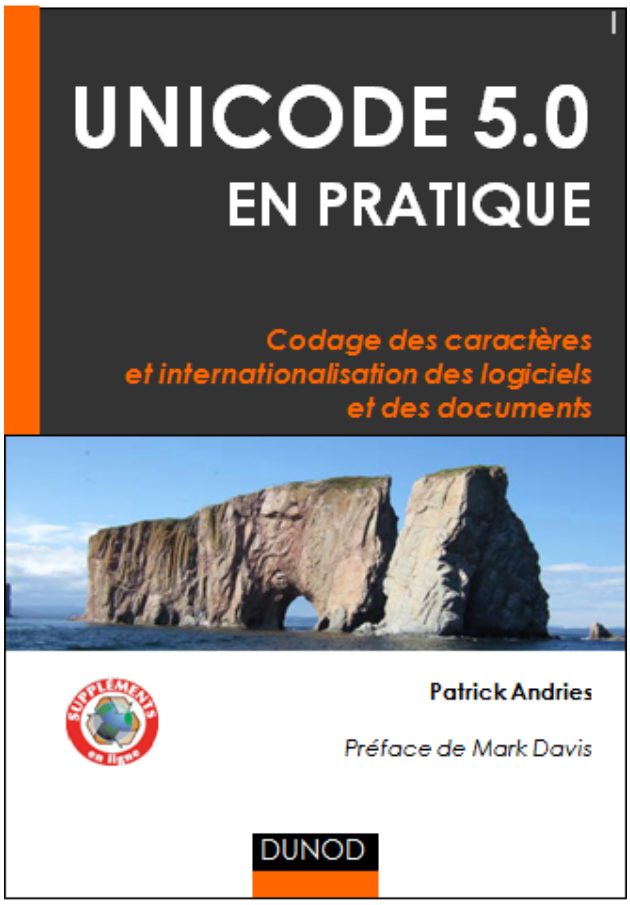
« Je ne connais pas d'autre ressource moderne, ni en anglais ni en français, regroupant une telle gamme de sujets utiles, et les expliquant de façon si claire et si accessible à un large public. J'espère que vous apprécierez cet ouvrage autant que moi. »

Mark Davis, Ph. D.
Président et cofondateur du Consortium Unicode

Commandez en ligne



Préface

Table des matières

Avant-propos

Premier Chapitre

Index

Errata

Entretien avec Ken Whistler, directeur technique du consortium Unicode

Patrick Andries

Conseils Hapax
2155, rue Lavoie
Longueuil (Québec) J3Y 8S7
hapax@iquebec.com
http://hapax.iquebec.com

RÉSUMÉ. Entretien avec Ken Whistler, directeur technique du consortium Unicode sur la naissance du standard Unicode, sa relation avec ISO/CEI 10646 et d'autres normes. Ken Whistler se penche sur certains problèmes de mise en œuvre et de normalisation des langues régionales européennes et africaines. Il répond aux inquiétudes d'aucuns quant à la baisse de qualité typographique qu'Unicode entraînerait. Il commente ensuite les imperfections et les incohérences perçues dans l'implantation d'Unicode : inclusion de glyphes, unification han partielle, complexité croissante, introduction de nouveaux caractères de formatage et formes de normalisation. Suit une discussion sur l'importance d'Unicode dans les bases de données et de l'avenir potentiel d'Unicode.

RÉSUMÉ EN CRI[1] ᐧᐄᓵᐧᐋ ᐋ ᐃᒋᔅᑖᓪ᙮ ᑳ ᐊᔨᒥᐦᐋᑭᓈᑦ ᑫᓐ ᐧᐃᓯᓚᔅ, ᑳ ᐧᐋᐃᓯᓈᐲᐦᑖᑦ ᒫᒨ ᑳ ᓂᔅᑐᔨᔅᑖᑭᓈᓪ ᐹᔨᑯᔨᓪ ᐋ ᐃᔑ ᒥᔨᓂᐦᐄᒫᐱᑭᓈᓪ ᐅᐦᒋ ᐊᓂᔮᔮ ᐋ ᐊᔨᐧᒫᐅᐦᐄᑯᑦ ᑳ ᐅᐦᒋ ᒋᐦᒋᐲᐦᑖᑭᓈᓪ ᐹᔨᑯᔨᓪ ᒫ ᐃᔑᓈᑯᐦᓪ ᐋ ᒥᔨᓂᐦᐄᒫᐱᑭᓈᓪ, ᑭᔮ ᐊᓐ ᐋ ᐃᔑ ᑖᑭᐦᐄᒫᐲᓪ ISO/CEI 10646 ᑭᔮ ᑯᑎᑊᐦ ᐋ ᒋᔅᒋᓅᐧᐋᐱᐦᑖᑭᓈᓪᐦ ᐋ ᐃᔑ ᒥᔨᓂᐦᐄᒫᐱᑭᓈᓪᐦ᙮ ᐊᓐ ᑳ ᓵᐳᒋᔅᒫᔨᐦᑎᑊ ᑫᓐ ᐧᐃᓯᓚᔅ, ᒌ ᒥᒋᓈᐱᐦᑎᒍ ᐊᓂᔮ ᐱᔅᓪ ᒫᐧᑳᔦ ᐋᑲ ᓂᐦᐄᐲᔨᓪ ᐹᔨᑯᔨᓪ ᐋᐧᐄ ᐃᔑ ᒥᔨᓂᐦᐄᒫᐱᑭᓈᔨᓪ ᐊᓂᑖ ᐋᑳᒥᒥᐦᒋᑭᒦᐦᓪ ᐊᓂᔮ ᓂᓈᐦᑰ ᐋ ᐃᔑ ᐊᔨᒥᓈᓈᔨᓪ ᑭᔮ ᐊᓂᑖ ᐋ ᐃᔑ ᒥᔨᓂᐦᐄᒫᓈᔨᓪ ᐧᐄᐹᔅᑎᑯᔑᔦ ᐊᔅᒌᐦᓪ᙮ ᑮᐧᐸ ᑳ ᓂᔅᐧᑳᐅᔅᑖᑦ ᐊᓂᔮ ᑳ ᐃᑖᔨᐦᑖᑭᓈᔨᓪ ᐋᑯᔅ ᐊᑎᑑ ᒫ ᓅᐦᑎᓂᑭᓈᔨᓪ ᐊᓂᔮ ᐋ ᐃᔑ ᒋᔅᒋᓅᐧᐋᐱᐦᑖᑭᓈᔨᓪ ᐋ ᒥᔨᓂᐦᐄᒫᐱᑭᓈᔨᓪ ᐊᓐ ᑳ ᐃᔑ ᓂᑐᐧᐋᔨᐦᑖᑭᓈᓪ ᒫ ᐃᔑᓈᑯᐦᓪ ᐹᔨᑯᔨᓪ ᐋ ᐃᔑ ᒥᔨᓂᐦᐄᒫᐱᑎᑭᓈᓪ᙮ ᐋᔦᐠ ᑭᔮ ᑳ ᐃᔑ ᐧᐄᐦᑎᑊ ᐊᓂᔮ ᐱᔅᓪ ᒫᐧᑳᔦ ᐋᑲ ᒫ ᓂᐦᐄᐲᔨᓪ ᒥᓅ ᐋᐱᒋᐦᑖᑭᓈᒫ ᐹᔨᑯᔨᓪ ᐋ ᐃᔑ ᒥᔨᓂᐦᐄᒫᐱᑎᑭᓈᓪ: ᒫ ᐊᔅᑎᓂᑭᓈᓪᐦ ᒥᔑᓈᐱᔅᑭᐦᐄᑭᓂᐦᑳᓂᔅ ᐃᔅᐱᔑ ᐧᐄ ᐊᔨᒍᐧᐃᓂᔅᐦ, ᐱᔅᓪ ᐋ ᐊᔅᒋᐲᓪ ᐦᐋᓐ ᐹᔨᑯᔨᓪ ᐋ ᐃᔑ ᒥᔨᓂᐦᐄᒫᓈᓪ, ᐋ ᐊᑎ ᒥᔑᐦᑖᐲᓪ ᒫᒫᐦᓪ ᐋ ᐃᔑ ᒥᔨᓂᐦᐄᒫᓈᓪ, ᐋ ᐧᐋᐱᐦᑏᐧᐋᓈᓪ ᐋ ᐅᔅᑳᓪ ᐊᔨᒍᐧᐃᓂᔅᐦ ᐋ ᐃᔑ ᐋᐱᑎᓪᐦ, ᐋ ᐧᐋᐱᐦᑏᐧᐋᓈᓪ ᑯᐃᔅᐠ ᒫ ᐃᔑ ᒥᔨᓂᐦᐄᒫᐲᓪ᙮ ᐋᐅᐠ ᐋ ᐊᔨᒍᑖᓪ ᐋ ᐃᔅᐱᔑ ᒋᔅᑎᒫᐅᐦᓪ ᐹᔨᑯᔨᓪ ᐋ ᐃᔑ ᒥᔨᓂᐦᐄᒫᐱᑎᑭᓈᓪ ᐊᓂᑖ ᒫ ᐃᐦᑎᑎᑭᓈᓪ ᑳ ᒥᔨᓂᐦᐄᒫᐲᓪ ᐋ ᒫᒨᔨᓈᑖᓪ ᒫᐧᑳᓐ ᑭᔮ ᐊᓐ ᐋ ᐃᔑ ᐱᑯᓵᔨᐦᑖᑭᓈᓪ ᒫ ᐃᔑᓈᑯᐦᓪ ᐹᔨᑯᔨᓪ ᐋ ᐃᔑ ᒥᔨᓂᐦᐄᒫᐱᑎᑭᓈᓪ ᐊᓂᑖ ᐃᔑ ᓅᔅᑖᒥᐦᓪ᙮

ABSTRACT. Interview with Dr. Ken Whistler, technical director of the Unicode Consortium on the origin of the Unicode standard and its relation to ISO/IEC 10646. Dr. Whistler answers several questions related to the perceived implementation problems and apparent weaknesses of the Unicode standard.

---

[1] Le cri, une des langues autochtones parlées au Québec, est pris en charge par Unicode.

Document numérique. Volume 6 – n° 3-4/2002, pages 1 à X

MOTS-CLÉS : Unicode, consortium, Whistler, ISO/CEI 10646, mise en œuvre, questions, base de données, formes normalisées, glyphe, jeux de caractères, CJC, unification han, tifinagh, breton.

ᐊᑭᒧ·ᐃᓐᐪ ᐊᑦ ᑎᖅᑲᓕᑦᑕᑯᓚᖅ: ᒪᒧ ᑲ ᓂᔅᑐᑭᓕᔅᑕᑯᓄᖅ ᐸᑭᔾᔪᑦ ᐊᑦ ᐃᓗ ᒥᓯᔾᓚᐃᑎᑯᓄᖅ, ·ᐃᓯᓕᔅ, ISO/CEI 10646, ᐊᑦ ᑎᖅᐱᐊᖅᑕᑯᓄᖅ, ·ᑲᑎᔅᓗᒧᐪᖅ, ᐊᑦ ᒪᒧᑭᓕᑕᒃ ᒪ·ᑲᖅ ᑲ ᒥᓯᔾᐃᓗᐊᖅ, ᔨᐃᔅᔪ ᐊᑦ ᐃᓗ ᒥᓯᔾᐃᓗᐊᖅ, ᒥᓗᓕᐱᔅᑭᑎᖅᐃᓐᓂᖅᑲᓂᖅ, ᐊᑦ ᒪᒧᑭᓕᑕᒃ ᐊᑦᑐᒧ·ᐃᓂᖅ, CJC, "ᐊᑭᐪ ᐸᑭᔾᔪᑦ ᐊᑦ ᐃᓗᓕᔨᑦᑕᑯᓄᖅ ᐊᑦ ᒥᓯᔾᐃᓗᐊᖅ, ᑎᐃᑦᐃᓐᓇ, ᐱᓪᑎᐪₓ

KEYWORDS: Unicode, Consortium, Whistler, ISO/IEC 10646, implementation issues, normalized forms, glyph, character sets, CJK, Han unification, Tifinagh, Breton.

PATRICK ANDRIES[2] — M. Whistler, quand avez-vous commencé à vous intéresser à Unicode et quel est votre rôle aujourd'hui au sein du consortium Unicode ?

KEN WHISTLER — J'ai obtenu mon doctorat en linguistique à l'Université de Californie à Berkeley en 1980. Après quelques années à l'université, je me suis retrouvé de plus en plus impliqué dans le développement de logiciels destinés aux linguistes simplifiant la création de polices, l'analyse linguistique et la rédaction de dictionnaires. Par la suite, l'aspect informatique m'a tellement intéressé que j'ai changé d'emploi et obtenu en 1989 un poste en internationalisation auprès d'une société de la Silicon Valley appelée Metaphor.

Metaphor était alors confronté aux problèmes liés à l'internationalisation de son important logiciel, j'étais un des informaticiens chargés de trouver une solution à l'utilisation des jeux de caractères. Mon patron chez Metaphor venait de Xerox, il y avait connu Joseph Becker et savait que ce dernier travaillait sur quelque chose qu'il appelait « Unicode ». Nous avons alors décidé de rencontrer M. Becker et quelques-uns de ses collègues pour discuter de la problématique liée aux caractères. Depuis lors, je travaille dans le domaine d'Unicode.

Cette implication précède la fondation du consortium Unicode par plus d'un an.

Voilà plusieurs années déjà que je suis le directeur technique du consortium Unicode. Ce poste signifie que je me concentre sur les problèmes techniques reliés au standard Unicode, plutôt que sur les aspects administratifs. J'ai également été un des principaux rédacteurs de chacune des versions du standard Unicode. Actuellement, je suis en plein dans la rédaction de la prochaine version du standard Unicode, la version 4.0.

PATRICK ANDRIES — Quand et comment Unicode a-t-il commencé ? S'agit-il surtout de l'initiative de quelques personnes passionnées ou d'une action concertée de la part des principaux fabricants de logiciels américains ?

KEN WHISTLER — Le codage réel d'Unicode débuta en 1987 quand deux idées issues du Centre de recherche de Xerox à Palo Alto (PARC) et d'Apple convergèrent. La première idée était la conception d'un codage de grande dimension à largeur fixe qui permettrait d'éviter les difficultés de traitement associées aux codages multioctets de

2 Les notes de bas de page sont de Patrick Andries.

l'époque. La seconde était l'« unification han » comme méthode permettant de représenter dans un seul jeu de caractères CJC étendus des textes chinois, japonais ou coréens. Pendant 1987 et 1988, Apple et Xerox travaillèrent à l'identification des caractères « chinois » de plusieurs jeux de caractères asiatiques et à la production de tableaux fondamentaux de transposition. Le résultat de ces travaux constitua l'amorce de notre projet de « simplement continuer à coder toutes les autres écritures du monde ».

Ce projet rayonna en dehors d'Apple et de Xerox quand, en 1989, des informaticiens de plusieurs sociétés, y compris moi-même chez Metaphor, ont commencé à se réunir régulièrement pour compléter et normaliser un jeu de caractère universel qui pourrait être utile à toute l'industrie logicielle. Le nombre de participants à ces réunions augmentant, l'association devint officielle et c'est ainsi que l'on fonda le consortium Unicode en janvier 1991.

Globalement, je pense que ce projet est né de la vision de quelques personnes, du moins au début. Les dirigeants de société n'ont pas l'habitude de penser à des choses comme les jeux de caractères ; ce sont les informaticiens responsables de l'internationalisation qui lancèrent le projet. Il ne s'agit donc pas d'une commande venant de la direction, mais d'une initiative de « la base » de plusieurs sociétés.

Toutefois, une fois que l'architecture choisie par le consortium Unicode a paru viable, les fabricants de logiciels prirent des décisions-clés qui intégrèrent Unicode dans leurs produits. Ce sont ces décisions, plus particulièrement de la part de Microsoft et d'IBM, qui ont été déterminantes dans le succès du standard Unicode.

PATRICK ANDRIES — Quelles sont, selon vous, les différences de structure et prises de décision entre le consortium et d'autres organismes comme le W3C et l'ISO ?

KEN WHISTLER — Le consortium est organisé de la même manière que de nombreux autres consortiums industriels. Des membres à part entière (dix-neuf à l'heure actuelle) cotisent et sont responsables des activités du consortium. Cependant, toutes les questions techniques, sont confiées au comité technique Unicode (CTU) qui se réunit quatre fois l'an. Les membres à part entière y envoient des délégués ayant droit de vote, toutefois ces réunions sont publiques. Les décisions et comptes rendus de ces réunions sont publiées sur le site internet du consortium.

Le standard Unicode est né de l'accumulation de décisions isolées – certaines concernent l'ajout de caractères, d'autres traitent de l'architecture ou de la documentation. Ces décisions isolées sont alors regroupées dans une version particulière du standard soumise tout d'abord à une période d'examen public. Ensuite la version définitive est soumise aux voix pour approbation finale avant sa publication.

Le fonctionnement du W3C ressemble à celui du consortium, si ce n'est qu'il rassemble plus de membres et que son mandat en matière de standardisation de l'internet est plus large. L'adhésion au W3C est nettement plus coûteuse que celle au consortium Unicode et l'accès aux travaux en cours du W3C dans de nombreux domaines est habituellement beaucoup plus restreint aux membres que ce n'est le cas pour les travaux du consortium

Unicode. Le W3C et le consortium Unicode font appel à leur site internet comme un mécanisme à part entière de participation dans l'élaboration des standards et la diffusion des standards approuvés.

L'ISO fonctionne quelque peu différemment en tant qu'organisme de normalisation. En effet, son travail prend place dans le cadre d'un traité international associé à l'OMC. Les membres des sous-comités associés au JTC1 (le Comité technique conjoint 1, responsable des normes informatiques) sont des organismes nationaux. Chaque nation qui désire participer accrédite un seul représentant. Dans le cas des États-Unis, par exemple, cet organisme national est l'INCITS, pour la France, c'est l'AFNOR.

Les normes internationales de l'ISO sont élaborées par des groupes de travail mis sur pied par les différents sous-comités techniques. Quand les projets de norme sont jugés prêts, ils passent par trois tours de scrutin national. Les commentaires des organismes nationaux issus des deux premiers tours sont incorporés dans le texte de la norme proposé pour le troisième tour. Si le projet est approuvé lors du troisième tour, on le publie alors sous la forme d'une norme. Le développement progressif de la norme s'effectue grâce à un processus d'amendements successifs.

PATRICK ANDRIES — Comment la norme ISO 10646 et le standard Unicode ont-ils fini par fusionner ?

KEN WHISTLER — Le tout fut une affaire assez décousue. Le projet ISO 10646 pour un codage de caractères universel remonte aux environs de 1984. Vers 1989, cependant, il devint clair que l'ISO 10646 sous sa forme proposée se conformerait à l'architecture sur 8 bits prédominante à l'époque[3]. Ce choix aurait cependant signifié que la 10646 aurait été une solution totalement inadéquate pour le traitement des caractères et qu'il ne se serait pas attaquer au problème d'utilisation des caractères CJC correctement. De nombreuses écritures auraient également été omises. En outre, la 10646 proposée avait totalement omis d'aborder les questions de compatibilité avec les nombreux jeux de caractères privés préexistants.

Pendant un certain temps, on eut l'impression que certains problèmes pourraient être résolus au sein du groupe de travail (JTC1/SC2/GT2), mais quelques-unes des décisions importantes furent renversées lors d'une réunion cruciale du SC2 en 1990. Retournement qui ne fit que mettre de l'huile sur le feu du mécontentement qui poussait au développement d'un concurrent, le standard Unicode.

L'apogée de la confrontation eut lieu en 1991. Le consortium Unicode venait d'être fondé et il publia son standard. Entre-temps, le SC2, résolu de publier sa norme, mit au voix l'ISO 10646 comme projet de norme internationale (« DIS »). Le scrutin à cette étape fut très animé, compte tenu de la publication d'Unicode. Le projet fut rejeté. Incapable de publier une norme internationale qui aurait écarté l'insignifiant standard Unicode, le SC2 accepta en définitive, bien que quelque peu à contrecœur, de rencontrer

3 L'ISO 10646 proposait un mécanisme de compatibilité, similaire à l'ISO/CEI 2022 qui aurait permis l'adressage des différents groupes de caractères grâce à des caractères d'échappement. Soixante-quatre positions de code étaient réservées pour les caractères de commande.

les principaux acteurs du consortium Unicode afin de trouver une façon de sortir du pétrin. Des deux côtés, les esprits les plus conciliants l'emportèrent et, en 1993, une version complètement révisée de l'ISO 10646 fut approuvée et publiée. En même temps, le consortium Unicode publia sa version 1.1, une refonte de sa version 1.0, afin de correspondre code par code à l'ISO/CEI 10646-1:1993.

PATRICK ANDRIES — Le consortium Unicode et le comité correspondant de l'ISO sont-ils pilotés par des sociétés américaines ? Trouvez-vous que les autres pays participent suffisamment ?

KEN WHISTLER — Si, de prime abord, il peut paraître que le consortium Unicode ne représente que les intérêts de sociétés et l'ISO les intérêts nationaux, cette division est dans la pratique nettement moins tranchée. Et dans le cas du standard Unicode, la situation est particulièrement complexe.

Le consortium Unicode n'élabore pas le standard Unicode en complète indépendance. Voilà plusieurs années déjà que l'élaboration du standard Unicode est synchronisée avec le travail effectué sur l'ISO/CEI 10646 par le JTC1/SC2/GT2, le groupe de travail responsable de cette norme internationale. Aucune addition n'est apportée à la norme ou au standard sans consultation mutuelle, car les deux parties sont convaincues qu'il vaut mieux, pour tous, de n'avoir qu'un seul codage universel, plutôt que deux (ou plus) qui seraient incompatibles.

En outre, la participation au CTU et au GT2 est croisée. Certaines nations ont décidé de ne pas participer au GT2, mais plutôt de prendre directement part aux délibérations du CTU concernant leurs problèmes de codage de caractères. Ainsi, les gouvernements de l'Inde et du Pakistan sont tous deux membres du CTU mais sont absents du GT2. Le CTU représente également les intérêts des bibliothèques universitaires[4] dans le codage des caractères. Au sein du GT2, certaines délégations nationales, comme les États-Unis, le Japon et l'Allemagne, représentent les intérêts de leurs entreprises et de leurs universités. D'autres pays collaborent également de manière assidue. La Chine, par exemple, est un participant particulièrement actif et tenace, elle représente ses intérêts et non ceux des sociétés américaines.

Et, ironiquement, l'organisation sur des bases nationales de l'ISO et les différents intérêts des organismes nationaux signifient souvent que les langues et les écritures minoritaires sont mieux représentées par le CTU que par l'ISO. Au sein de l'ISO, il arrive qu'ils soient représentés par procuration, pour ainsi dire, par certains organismes nationaux, comme celui de l'Irlande, qui adoptent un point de vue universaliste dans la défense de leurs intérêts.

Malheureusement, la France a décidé de ne participer activement ni au sein du WG2 ni dans le travail du CTU. Cela explique peut-être la perception que certains Français peuvent avoir au sujet du standard Unicode. En fait, le Canada a été nettement plus actif, avec une forte participation du Québec[5], en particulier. Et vos lecteurs seront

4 La plupart par le truchement du RLG, un consortium américain de bibliothèques de recherche.

5 Dans la personne d'Alain Labonté du Conseil du Trésor, <alb@sct.gouv.qc.ca>.

peut-être intéressés d'apprendre que le rédacteur actuel de l'ISO/CEI 10646 se nomme Michel Suignard, un bretonnant ! Il est également un des rédacteurs du standard Unicode.

PATRICK ANDRIES — Le consortium Unicode ne publie-t-il que le standard Unicode ?

KEN WHISTLER — À l'origine, la mission du consortium Unicode était d'élaborer et de tenir à jour le standard Unicode. Mais, depuis quelques années, ses membres ont décidé que la mise en œuvre d'Unicode nécessitait de se pencher sur d'autres problèmes reliés aux caractères. Pour certains d'entre eux, le Comité technique Unicode avait les compétences requises pour produire le type de standard nécessaire. C'est ainsi que le CTU a officiellement publié une standard séparé sur la compression de textes Unicode et une autre (étroitement aligné sur l'ISO/CEI 14651) sur le tri de chaînes Unicode. Il a également publié un standard sur les formes normalisées de textes Unicode, mais celui-ci a été officiellement incorporé dans le standard Unicode.

PATRICK ANDRIES — Quelle est selon vous la plus importante contribution d'Unicode ?

KEN WHISTLER — Assurément la création d'un codage universel permettant la transmission d'un bout à l'autre du monde de textes de manière sûre. Son impact sera sensible sur l'informatique et les communications du XXI^{e} siècle.

La métaphore de la Tour de Babel s'avère appropriée. Vers la fin du siècle dernier, un grand nombre de petits (et grands) jeux de caractères proliférait, avec un très grand nombre de manières, non compatibles, de représenter des caractères. Le volume croissant de textes représentés et stockés dans les ordinateurs et les bases de données aurait très bien pu signifier que les systèmes informatiques n'auraient plus été capables d'échanger des données, chacun replié sur son propre codage.

Unicode est un élément-clé qui annonce un réseau de données évolutif, omniprésent et disponible. Avec internet (ou un de ses successeurs plus puissants) dans le rôle du réseau et le web dans celui de la bibliothèque et de l'éditeur planétaires, Unicode rend cette information disponibles dans tous les systèmes humains d'écriture.

PATRICK ANDRIES — Certaines écritures africaines représentant des populations importantes (le tifinagh[6] par exemple) ne sont pas aussi bien représentées que des écritures américaines bien moins utilisées (le syllabaire canadien, le chéroki et le déséret[7], par exemple). Pourquoi ? Que faire pour corriger cette situation ?

KEN WHISTLER — Cette situation reflète les intérêts des participants les plus enthousiastes par rapport au processus de normalisation et, parfois, tout simplement, la complexité inhérente à la normalisation de certaines écritures.

Dans le cas du chéroki et du déséret, par exemple, leur normalisation ne présentait aucun obstacle technique notable. Ces deux écritures alphabétiques possèdent peu de

6 L'écriture originale des Berbères.

7 Pour ces autres écritures, se référer au texte du standard <http://hapax.iquebec.com>. Le résumé de la première page de cet entretien est en cri, il est écrit à l'aide du syllabaire canadien.

signes, sont bien définies et stables depuis plus d'un siècle. Ces systèmes d'écriture furent justement choisis tôt dans le processus de normalisation parce qu'ils sont simples et possèdent un répertoire limité de signes ; ajoutons qu'il nous était également facile de trouver des experts.

Il en va tout autrement pour une écriture comme le cunéiforme suméro-akkadien, en revanche. Son répertoire de signes est étendu. Il s'agit également d'une écriture antique dont nous possédons certes de nombreuses inscriptions, mais elles sont souvent fragmentaires et posent de nombreux problèmes d'interprétation. Il faut du temps à des philologues et d'autres spécialistes pour aboutir à une solution consensuelle sur la manière de normaliser le codage de cette écriture.

Le tifinagh est un cas intéressant. Il s'agit d'une écriture assez répandue en Afrique du Nord-Ouest, mais elle n'a pas de forme nationale normalisée et aucun pays[8] ne représente ses locuteurs dans le processus normatif (il n'existe pas d'État-nation berbère). Il existe, en outre, un grand nombre de variantes locales et historiques et quelques caractères à l'identité problématique. Habituellement, dans le cas d'écritures nationales, l'État établit une norme scolaire et définit un jeu de caractères bien défini qui peut alors être intégré dans Unicode. Pour les écritures minoritaires, la normalisation des caractères peut impliquer un long travail de recherche et des opinions contradictoires quant à la voie à suivre. En fait, le GT2 et le comité technique Unicode étudient l'inclusion du tifinagh dans l'ISO 10646 et Unicode[9].

Des discussions avec des organisations berbères sont en cours, mais il reste quelques difficultés à résoudre quant à la manière d'unifier certaines formes de l'écriture. Il reste encore du travail à faire.

Réunir les experts de différentes écritures, les usagers – qui peuvent regrouper des gouvernements ou des ONG intéressés par la renaissance ou l'utilisation d'une écriture autochtone – et les experts en normalisation de jeux de caractères et s'assurer que tous les bénévoles qui participent au projet aient le temps nécessaire pour effectuer le travail requis, voilà les véritables obstacles.

PATRICK ANDRIES — Des soucis similaires existent également en ce qui concerne les langues régionales européennes. Certains Bretons se plaignent de l'absence du K barré (Ƙ) et de caractères uniques pour représenter CH et C'H.

KEN WHISTLER — Pour ce qui est des caractères « manquants » comme le CH et C'H en breton, il s'agit d'une méprise quant à l'objet des codages ISO/CEI 10646 et Unicode. En effet, ces normes codent les caractères d'*une écriture* et non d'un alphabet. Il existe

8 Il s'agirait là d'une belle tâche pour les institutions de la Francophonie et les nations (France et Canada (Québec)) qui siègent au comité de l'ISO sur les jeux de caractères : représenter les intérêts linguistiques et informatiques des pays de la Francophonie. Un intérêt pour le khmer (et un dialogue plus prompt avec les experts français ou cambodgiens de cette écriture) aurait pu éviter bien des quiproquos et désagréments.

9 Voir la proposition préliminaire (en anglais uniquement, malheureusement) <http://std.dkuug.dk/jtc1/sc2/wg2/docs/n1757.pdf>.

des milliers de langues qui utilisent l'écriture latine, chacune a son propre alphabet. Mais le standard Unicode ne code qu'une seule fois l'écriture latine, pour toutes ces langues ; il ne code pas l'écriture latine autant de fois qu'il existe de langues l'utilisant.

Il arrive souvent qu'un alphabet latin utilise des digrammes – deux lettres qui ne représentent qu'un son de base de la langue. Mais, pour un codage universel de caractères comme Unicode, on ne peut coder toutes les combinaisons de digrammes, trigrammes ou polygrammes ; il y en a tout simplement trop. De toute façon, coder les digrammes en tant que caractères autonomes compliquerait de nombreux traitements textuels courants. Le cas du « ch » est bien connu. Cette combinaison est, après tout, un digramme en espagnol, en slovaque (avec un son complètement différent) et même en français ! L'espagnol et le slovaque le considèrent comme une « lettre » de leur alphabet, mais pas le français.

Un examen attentif du standard Unicode révèlera, bien sûr, l'inclusion de digrammes (plusieurs pour le croate et un pour le néerlandais). Mais il s'agit d'exceptions et non de la règle. Chaque exception se justifie : le plus souvent par la compatibilité avec un jeu de caractères source. Le CTU décourage le codage de nouveaux digrammes ; les quelques-uns présents compliquent déjà certaines opérations sur les textes.

Le K barré du breton est un cas particulier, apparemment. Il s'agit d'une ligature représentant le préfixe « Ker ». Il existe, évidemment, des milliers d'abréviations manuscrites ou d'origine manuscrite pour l'ensemble de toutes les écritures du monde, y compris des centaines, au bas mot, pour l'écriture latine et cela rien qu'en Europe. La manière de traiter les abréviations manuscrites dans les textes informatisés est de les écrire en toutes lettres – c'est une pratique courante dans les systèmes bibliographiques, pour les signes d'abréviation latins, car il permet d'obtenir des résultats nettement plus fiables lors d'une recherche documentaire. Quand il est possible de représenter une abréviation à l'aide de caractères ordinaires, comme dans le cas du K̶, cette méthode ne présente pas vraiment de problème. Quand la forme ligaturée de l'abréviation s'avère nécessaire à l'affichage, la meilleure manière de l'obtenir est d'utiliser des *polices* conçues pour ce faire, et non d'essayer de coder toutes les ligatures au sein du jeu de caractères.

Une fois encore, un examen attentif du standard Unicode révèlera la présence de certaines ligatures au sein de celui-ci – quelques-unes pour l'écriture latine mais un grand nombre pour l'écriture arabe. Mais toutes furent codées pour des raisons de compatibilité très précises et circonscrites. L'utilisation de ces ligatures pour représenter des textes courants n'est pas recommandée et il n'y a aucun doute que le CTU ne voit pas d'un bon œil le codage de nouvelles ligatures.

PATRICK ANDRIES — Ainsi, même si le Breton ne connaît pas la lettre C et le CH n'est pas un digramme en breton, le CH est considéré un digramme dans l'écriture latine. Quels sont les avantages de cette unification par écriture plutôt que par langue écrite ?

KEN WHISTLER — Bien, quand on parle digramme il faut garder à l'esprit les différents niveaux en présence : les phonèmes d'une langue, les graphèmes d'un système

d'écriture, les lettres d'un alphabet et les caractères d'une écriture commune à plusieurs langues. Toutes ces notions ne s'emboîtent pas nécessairement parfaitement, chose dont on se rend compte quand on essaye de découvrir les règles orthographiques. Dans un bon système d'écriture phonémique, les unités de base utilisées, les graphèmes, correspondent étroitement aux phonèmes. Et typiquement, si un tel système d'écriture est bien conçu, il aura un alphabet aux lettres simples qui correspondent bien aux graphèmes.

Considérer le codage dans une perspective liée à l'*écriture* en question plutôt que la langue présente l'avantage de permettre à un seul codage de représenter des textes de milliers de langues utilisant cette écriture (dans le cas latin). Pour prendre un exemple quelque peu extrême, il existe une langue d'Afrique australe du nom de *!Xóõ* qui possède de nombreux phonèmes complexes écrits à l'aide de polygrammes dans le système orthographique latin. Un de ces phonèmes, un clic coarticulé, s'écrit *dts'kx'*. Il est aisé de transcrire ce phonème en Unicode, en fait on peut également le représenter en Latin-1 ou même en ASCII. Si on venait à insister sur l'existence d'un caractère autonome *dts'kx'* reflétant le phonème *!Xóõ*, il faudrait que cette lettre soit d'abord codée dans Unicode avant de pouvoir représenter correctement le *!Xóõ*, processus qui pourrait prendre plusieurs années.

Remarquez que la méthode traditionnelle de codage des digrammes (et polygrammes) n'a pas été inventée par Unicode. C'est, *grosso modo*, la méthode que tous les jeux de caractères utilisent avec de très rares exceptions. Il est tout simplement plus efficace de coder les textes au niveau de l'écriture (en tant que lettres latines) et de laisser les traitements propres à la langue, comme le tri, prendre en compte la valeur alphabétique de ces combinaisons.

PATRICK ANDRIES — Certains se plaignent du fait que leur écriture n'est pour l'instant prise en charge qu'à l'aide de suites décomposées de caractères combinatoires et que cela rend leur mise en œuvre plus difficile car cette composition dynamique ne fonctionne pas toujours correctement aujourd'hui. L'inclusion de caractères précomposés pour ces écritures n'améliorerait-il pas leur prise en charge ?

KEN WHISTLER — L'intégration de nouveaux caractères précomposés pour d'autres alphabets latins pourrait simplifier *certains* types de mises en œuvre logicielles, mais il nous faut un peu de perspective ici. Tout d'abord, il existe tout simplement trop de combinaisons précomposées pour les milliers de langues qui utilisent l'écriture latine. Nous passerions notre temps à définir toutes les combinaisons employées, alors que toutes les lettres de base et tous les diacritiques sont déjà définis dans Unicode. Insister sur l'utilisation de caractères précomposés ne ferait que retarder la prise en charge des langues concernées car elles devraient attendre l'ajout de ces lettres précomposées dans Unicode.

En outre, à ce stade dans le développement d'Unicode, l'intégration de caractères latins précomposés supplémentaires ne changerait à peu près rien au niveau de la mise en œuvre. En effet, les formes de normalisation que l'internet, de nombreuses bases de données et autres systèmes imposent aujourd'hui décomposent tout simplement toutes

les formes précomposées afin d'éviter toute corruption des données de texte normalisées.

Pour éviter les restrictions potentielles liées à la composition dynamique (au fur et à mesure de laquelle les différents glyphes des caractères de base et des diacritiques doivent être placés par un moteur de rendu), il faut que les utilisateurs insistent plutôt sur des polices qui comprennent des *glyphes* précomposés. Les polices de caractères récentes permettent de dessiner à l'avance des glyphes qui correspondent à des formes complexes ; ces glyphes peuvent également être associés à une suite de caractères dans le texte. C'est exactement comme cela que les polices traitent habituellement les ligatures. On peut utiliser la même méthode pour traiter les combinaisons *lettres + diacritiques*.

Cette méthode n'a vraiment rien d'extraordinaire quand on considère Unicode dans son ensemble. Le rendu exact de la dévanâgarî, par exemple, nécessite un grand nombre de « conjointes » et d'autres formes spéciales de glyphes. L'ajout, à des polices dévanâgarî, des glyphes et des combinaisons de glyphes (pour les conjointes) nécessaires et non l'intégration, à Unicode, d'autant de nouveaux caractères permet de résoudre correctement ce problème. L'ajout de formes précomposées compliquerait considérablement la représentation sous-jacente des textes dévanâgarî et rendrait *nettement* plus difficile l'écriture de bons logiciels pour le traitement ou l'affichage du dévanâgarî.

Quand on songe au standard Unicode, il ne faut pas se faire piéger en pensant qu'un caractère stocké corresponde à un glyphe à l'affichage, et ceci est valable autant pour l'écriture latine que les autres écritures que l'on considère la plupart du temps comme complexes car exotiques.

PATRICK ANDRIES — Dans le même ordre d'idée, d'aucuns reprochent à Unicode d'avoir apparemment favorisé certaines écritures en les codant d'une façon plus simple ou plus adéquate. Le gothique, par exemple, n'est pas codé en tant que tel et son affichage correct nécessite l'ajout d'un protocole de niveau supérieur ou de caractères de commande à sa translittération latine. Le khmer est un autre exemple où Unicode semble avoir voulu faire rentrer cette écriture dans le moule brahmi alors que la pratique courante aurait voulu qu'on utilisât une solution proche de celle choisie par Unicode pour le thaï[10].

KEN WHISTLER — Le standard Unicode *ne* vise explicitement *pas* à reproduire fidèlement les textes affichés. Il s'agit d'un standard codant des textes *bruts*. Unicode représente explicitement les caractères des différents systèmes d'écriture car il le faut bien pour représenter correctement des textes. Mais il faut bien établir la différence

10 On se rappellera que le Comité cambodgien pour la normalisation des caractères khmers en informatique avisait l'ISO le 14 mai 2002 qu'il adopterait à contrecœur la solution proposée par l'ISO 10646/Unicode pour coder leur écriture. Cette décision, écrivait le comité, avait été essentiellement prise parce que le Cambodge ne pouvait se permettre ni un retard supplémentaire dans la mise en œuvre de son écriture ni l'apparition d'un codage superflu et incompatible qu'entraînerait l'adoption d'une norme nationale rivale.

entre ce codage de base et celui qui consisterait à coder séparément tous les *styles* d'une écriture. Il existe ainsi au moins sept variantes stylistiques courantes pour les caractères chinois et des dizaines – voire des centaines – de styles de fantaisie. Si chacun de ces *styles* d'écriture han devait être codé séparément, en comptant 70 000 caractères chinois par style, plus d'un demi-million de caractères seraient bientôt réservés à cette seule fin, sans aucun avantage mais avec de nombreux désavantages.

De même, l'écriture latine s'écrit à l'aide de nombreux styles, certains fréquemment utilisés de nos jours (l'italique par exemple) alors que d'autres sont historiques ou régionaux (gothique et gaélique). Les experts en matières de codage sont généralement d'avis qu'il vaut mieux ne pas coder ces variantes stylistiques. Bien que l'on eût pu (apparemment) simplifier certains processus de rendu en codant plus de variantes stylistiques, cela n'aurait pas été sans compliquer bien d'autres processus. Ainsi, le codage séparé du gothique aurait compliqué la recherche de textes latins, car les requêtes qui fonctionnent en utilisant des caractères latins ne trouveraient pas les textes écrits en gothique – contre toute attente. De toute façon, la meilleure manière de traiter certaines singularités du gothique, comme la coupure de mots, est d'utiliser des logiciels adaptés à la langue concernée et non de coder plus de caractères.

Il est vrai que le codage du khmer a récemment soulevé une vive polémique. Toutefois, dans ce cas précis, il semble que le problème se résume à deux options techniques foncièrement équivalentes permettant de représenter les consonnes souscrites de cette écriture. L'avis des experts, aussi bien au Cambodge qu'à l'étranger, était partagé. Au bout du compte, la solution choisie par Unicode n'a pas la préférence exprimée par le groupe normatif cambodgien, bien que ce groupe admette que la proposition du consortium résout le problème en question. S'il faut retenir une leçon de cette controverse, c'est qu'il est impératif que toutes les parties concernées se rencontrent et communiquent aussi tôt que possible lors d'élaboration d'un codage afin de dégager un consensus avant qu'il ne soit trop tard pour remettre en question une décision désormais inscrite dans une norme.

PATRICK ANDRIES — Vous aviez mentionné plus tôt la corruption potentielle des données textuelles normalisées. Pourriez-vous développer ? Quelle importance les formes de normalisation Unicode ont-elles pour les bases de données ou les systèmes de gestion de documents ?

KEN WHISTLER — Les formes de normalisation Unicode jouent un rôle primordial dans les bases de données et d'autres systèmes de gestion de documents. En effet, les bases de données possèdent des index utilisés pour accélérer grandement l'accès à ces données. Pour être valables, ces index doivent renvoyer à des données bien formées ; ceci signifie entre autres que des chaînes équivalentes (en termes de normalisation des données[11]) doivent apparaître au même endroit dans l'index. Les requêtes structurées d'un SGBD sont également optimisées à l'aide d'algorithmes complexes afin de regrouper plus rapidement des données provenant de dizaines, parfois des centaines, de tables. Ces optimisations supposent également que les données sont bien formées. Les

11 Ainsi un « é », par exemple, doit-il être considéré égal à la suite de caractères « e + ´ ».

formes de normalisation sont le mécanisme qui permet à ces méthodes de fonctionner correctement.

PATRICK ANDRIES — Est-ce que l'ajout des seizets d'indirection, des formes normalisées, du tri Unicode, des sélecteurs de variantes et de caractères de commande n'a pas rendu Unicode trop compliqué par rapport à ce qu'aurait pu être un jeu de caractères classique ?

KEN WHISTLER — Il faut prendre en compte l'ampleur du problème que le standard Unicode essaie de résoudre : rien de moins que de coder les caractères de *tous* les systèmes d'écritures du monde, modernes et anciens. Il suffit de brièvement parcourir un des ouvrages classiques consacrés aux écritures du monde pour se rendre compte qu'établir un codage adéquat de tous ces caractères ne peut être une chose aisée. Ajoutons les complications inévitables quand on tente de faire fonctionner, de façon intégrée, tous ces caractères sur des ordinateurs récents. De nombreuses additions au standard découlent, presque inexorablement, de la nécessité de le mettre en œuvre. Les seizets d'indirection sont nés du besoin de disposer de plus de 64K caractères et d'avoir, pour des raisons de mises en œuvre, des formes de codage des caractères sur 8, 16 et 32 bits. Les formes de normalisation découlent de la nécessité de pouvoir comparer de façon fiable des chaînes binaires. Certains des nouveaux caractères de commande, comme les commandes de formatage bidirectionnel, ont pour origine l'obligation de pouvoir définir correctement les textes bidirectionnels, de sorte que les caractères arabes qu'un utilisateur envoie à un autre utilisateur puissent être interprétés de la même manière par l'expéditeur et le destinataire, sans quoi le codage Unicode ne serait d'aucune utilité dans l'échange de données. Et ainsi de suite.

Enfin, le désir légitime des personnes chargées de la mise en application d'Unicode d'avoir à leur disposition des solutions communes permettant de trier des données Unicode – une question fort complexe et ardue quand on considère un codage de caractères universel – explique l'intérêt que le consortium a porté à cet aspect. Pourquoi avoir 27 solutions, toutes défectueuses et différentes, alors qu'une méthode souple mieux conçue et standardisée peut être définie ? L'algorithme de tri Unicode permet de définir des préférences d'ordonnancement locales ; l'essence même de cet algorithme est de permettre sa personnalisation afin de respecter l'ordre de tri traditionnel d'une culture ou d'une langue particulière (par exemple, le français) sans devoir se soucier comment trier en même temps tous les autres caractères (arabes, chinois, éthiopiens, etc.).

Bref, à mon avis, le standard Unicode est juste assez compliqué pour fonctionner.

PATRICK ANDRIES — Il semble que certaines « erreurs » se soient glissées dans Unicode. L'inclusion de nombreux *dingbats* et d'étoiles Zapf vient à l'esprit. Le principe « une fois codé, éternel » n'est-il pas foncièrement malsain ? L'entropie ou la complexité ne risque-t-elle pas d'augmenter sans cesse ?

KEN WHISTLER — Si la question est de savoir si certains caractères aujourd'hui codés deviendront désuets à l'avenir, la réponse est, bien sûr, oui. De fait, le comité technique

Unicode et le GT2, à la suite de demandes de codage de caractères, ont inclu à dessein des caractères qui étaient déjà désuets au moment de les normaliser – tout simplement parce que les gens en ont besoin pour représenter correctement des textes anciens.

De nombreux anciens jeux de caractères ont été intégrés dans Unicode pour des raisons d'universalité. C'est une mesure de transition nécessaire pour s'assurer que le standard Unicode ne constitue pas un obstacle au passage des anciennes formes de stockage aux nouvelles. Dans le cas des *dingbats* Zapf, le standard reconnaît pleinement que ces caractères s'écartent, au moins en partie, des principes de codage qui distingue les caractères des glyphes. Toutefois, sur le terrain dans les années 90, de nombreux systèmes informatiques traitaient les symboles PostScript et les *dingbats* Zapf comme des *caractères* et, pour assurer leur transition vers Unicode, il a fallu prévoir des points de code correspondants dans Unicode.

Quant au « une fois codé, éternel », c'est le principe même de toute normalisation. Une norme n'est pas une norme fiable si ceux qui la tiennent à jour commencent à éliminer des caractères parce qu'ils pensent qu'ils sont bizarres ou inutiles. Ce genre de comportement aboutirait vite à l'abandon de cette norme.

Ce qui pourrait se produire, en revanche, c'est que des parties moins utiles d'Unicode destinée à une phase transitoire soient annotées et qu'on suggère de les éviter, sauf dans les mises en œuvre destinées à d'anciens systèmes. Ceci s'est déjà produit : ainsi aucune mise en œuvre Unicode de l'arabe ne se préoccupe du grand nombre de ligatures arabes codées dans Unicode, ligatures dont l'inclusion a constitué une erreur. Les implantations arabes correctes utilisent les caractères arabes de base et étendus et délèguent la formation des ligatures aux polices, comme il se doit.

PATRICK ANDRIES — D'aucuns se plaignent que les noms de l'ISO 10646[12] et d'Unicode portent des noms déroutants[13] ou des noms qui décrivent leur apparence plutôt que leur fonction (l'API par exemple). Pourquoi ces noms ne peuvent-ils être changés ?

KEN WHISTLER — Les noms de caractères ne sont que des identificateurs stables pour les caractères[14]. Nous avons appris à nos dépens que la « correction » des noms ne fait qu'embrouiller encore plus les choses et pose encore plus de problèmes dans les mises en œuvre, sans oublier les discussions interminables au sien des comités entre les personnes qui privilégient une correction par rapport à une autre ou ceux qui essaient de résoudre ce qui est au fond un problème de codage par une modification terminologique.

---

12 Tous les caractères portent également des noms français officiels (l'ISO/CEI 10646 a été officiellement publiée en français). Ces noms annotés sont disponibles à l'adresse suivante : <http://iquebec.ifrance.com/hapax/Tableau_annote.htm>.

13 Un exemple classique, dans les noms anglais, est la série des noms APL où U+22A4 ⊤ est un « down tack », mais U+2351 ⍑ est un « up tack overbar »…

14 Le numéro de caractère (U+2001) est un autre identifiant stable, bien que pour les humains moins mnémotechnique.

Il faut comprendre qu'il est impossible d'obtenir une liste de noms parfaite pour l'ensemble des caractères (et les milliers de symboles) du monde. Tenter d'obtenir une telle liste est une chimère, et les comités se sont mis d'accord sur l'attribution de noms adéquats, uniques et stables.

Certaines erreurs dans les noms de caractères crèvent les yeux, c'est vrai. Le standard Unicode annote ces caractères avec soin, afin d'assurer l'identification des caractères en question. Pour une norme, l'existence d'une erreur reconnue, bien annotée mais stable, est de loin préférable à un nombre aléatoire de corrections terminologiques qui pourrait sans cesse varier selon les caprices des changements proposés chaque année.

Remarquons que, même si les noms de caractères sont normatifs pour la norme, cela veut simplement dire qu'ils font partie de la partie stable de la norme et qu'ils jouent leur rôle d'identificateur de caractère. Cela ne signifie pas qu'on puisse en déduire d'autres informations normatives quant à l'utilisation du caractère ou que la norme force d'autres gens à appeler les caractères de la sorte. Ainsi, la traduction des noms de caractères est-elle permise et il n'y a pas de raison pour que des erreurs dans les noms de caractères ne soient pas corrigées lors de cette traduction[15].

D'ailleurs, le fait que U+002E porte le nom normatif en anglais de FULL STOP (une forme britannique) n'empêchera pas les locuteurs américains de l'appeler une « period » ou que d'autres disent en anglais « dot » quand ils épellent un URL.

PATRICK ANDRIES — Pourquoi l'ISO 10646 et Unicode comprennent-ils tant de caractères « préexistants » qui semblent enfreindre son modèle caractère-glyphe ?

KEN WHISTLER — C'était une des conditions de base pour garantir le succès d'Unicode comme codage qui remplacerait les jeux de caractères historiques. Les comités devaient s'assurer qu'Unicode pourrait jouer le rôle de pivot entre les jeux de caractères historiques de sorte que lorsque des logiciels Unicode échangeraient des données avec des systèmes ou des mémoires de données préexistants – ce qui devrait encore se produire pendant des décennies – aucune corruption de données ne se produirait par un défaut dans le codage-pivot.

Ce but impose, bien sûr, des choix de conception parfois contraires à ceux que l'on aurait arrêté si on avait conçu Unicode à partir de zéro, sans se soucier de pouvoir échanger des données avec des systèmes historiques. En fait, si on étudie avec soin le standard Unicode, les écritures récemment codées avec peu d'histoire informatique sont celles qui ont posé le moins de problème et bénéficient des codages les plus élégants. La kyrielle d'impuretés qui encombre le standard Unicode est reliée aux écritures dont la mise en œuvre est la plus ancienne. L'écriture latine est à cet égard de loin la pire contrevenante, suivie de près par l'arabe et les différents caractères de compatibilité

15 C'est le cas dans la traduction française. Le caractère ℘ porte en anglais le nom de SCRIPT CAPITAL P alors qu'il s'agit d'une minuscule… La version française de l'ISO/CEI 10646 nomme plus opportunément ce caractère FONCTION ELLIPTIQUE DE WEIERSTRASS.

hérités des différents jeux de caractères préexistants d'Extrême-Orient. En revanche, le singhalais et l'écriture éthiopienne[16] sont des modèles de simplicité et de cohérence !

PATRICK ANDRIES — Pourquoi Unicode comprend-il des caractères de formatage (bidi ou ligature) ? Est-ce que ceci ne devrait pas être laissé à des protocoles de niveau supérieur ? La possibilité de commander le formatage à deux niveaux (XML et Unicode pour le bidi) semble complexifier inutilement ces processus et leur ôter une certaine orthogonalité. Unicode pourrait-il déconseiller l'usage de ses caractères de commande à l'avenir ?

KEN WHISTLER — La raison principale pour laquelle Unicode comprend ces différents caractères de commande de formatage est de permettre la désambiguïsation des textes bruts dans des contextes où des protocoles de niveau supérieur pourraient être absents. Le cas bidirectionnel est un bon exemple. On peut faire un bon bout de chemin en n'utilisant qu'un simple algorithme bidirectionnel implicite qui se base sur les propriétés directionnelles des caractères présents dans le texte à formater, sans introduire parmi ces caractères de commandes de formatage particuliers. Mais il existe toujours des cas spéciaux (numéros de pièces, de téléphone et certaines inclusions directionnelles) où les règles implicites ne suffisent pas. Dans ces cas, il faut pouvoir imposer un affichage particulier à l'aide de commandes locales. Sans celles-ci un texte saisi pourrait ne pas s'afficher de la manière voulue. Ceci se produit souvent en l'absence de protocole de niveau supérieur de rendu directionnel : les formulaires de saisie de données ou d'autres objets de saisie simples qui n'acceptent et n'affichent que des textes bruts.

Évidemment, les choses peuvent se corser quand des commandes de formatage bidirectionnelles se retrouvent en concurrence avec des balises de formatage directionnelle, comme en HTML ou XML. Mais plutôt que d'insister pour que cette information ne soit codée qu'à un seul niveau, nous avons préféré élaborer des lignes directrices officielles qui définissent l'utilisation ou non de commandes de formatage bidirectionnelle (et d'autres commandes de formatage spécialisées) en présence de balises ou d'autres protocoles de niveau supérieur de formatage.

Pour ce qui est de déconseiller ces caractères de commande de formatage à l'avenir, j'aimerais mentionner que certains caractères de commande Unicode ont été déconseillés *dès leur inclusion*. Ils avaient été ajoutés au début de la fusion des répertoires Unicode et ISO/CEI 10646 pour des raisons que les membres du CTU trouvaient peu judicieuses. Il est possible que d'autres caractères de commande soient déconseillés à l'avenir, mais le CTU n'inclut de nouveaux caractères de commande qu'à la suite d'un examen critique, car ils peuvent bien sûr avoir un impact structurel sur le texte et causer des interférences inattendues avec des niveaux de la représentation textuelle autres que les caractères.

PATRICK ANDRIES — Est-ce que les formes de normalisation n'ont pas, en fin de compte, été introduites pour remédier à des faiblesses observées dans des versions

16 Parfois appelée amharique.

précédentes d'Unicode (introduction de caractères précomposés et décomposés, inclusion de plusieurs variantes du même caractère) ?

KEN WHISTLER — Bien sûr, mais que pouvait-on faire d'autre ? Afin d'assurer la compatibilité avec les jeux de caractères préexistants, le standard Unicode a dû inclure toute sorte de bizarreries, y compris des doublons et des erreurs de codage héritées des anciens jeux de caractères. Cela posait déjà un problème de normalisation des formes concurrentes.

Et pour réussir en tant que standard, Unicode devait inclure les caractères précomposés afin d'assurer une conversion aller-retour vers un certain nombre de codages historiques, notamment l'ISO 8859-1 Mais, pour coder complètement l'écriture latine, il lui fallait également coder les diacritiques. Cela fut bien sûr l'objet de longs débats, d'interminables débats diraient certains, au tout début, avant même qu'Unicode 1.0 ne soit publié. Mais certaines personnes très intelligentes et expertes ès normes de codage de caractères ne voyaient pas d'autre solution qui *primo* fonctionnerait et *secundo* s'imposerait. Cela posait un autre problème de normalisation de formes concurrentes.

La véritable erreur aurait été de ne pas affronter ces problèmes de normalisation des formes concurrentes et de ne pas définir un algorithme qui permet de résoudre ce problème.

PATRICK ANDRIES — La popularité et la diffusion d'Unicode ne pourraient-elles pas être responsables pour une baisse de qualité paradoxale dans la production typographique ? Avec l'adaptation des polices à Unicode et la reproduction des glyphes de référence qu'Unicode prévoit, certains craignent que des typographes conçoivent des polices inélégantes pour des écritures qu'ils ne maîtrisent pas. Ne craignez-vous pas qu'Unicode pourrait uniformiser vers le bas la typographie de notre planète ?

KEN WHISTLER — C'est à mon avis peu probable. Les normes typographiques continueront à être établies par des typographes professionnels qui ne se sentent pas tenus de respecter les tableaux de caractères de référence assez quelconques publiés par le consortium Unicode.

Rappelons que le standard Unicode et son pendant ISO, l'ISO 10646, sont des normes de codage de *caractères*. Leur domaine n'englobe pas la définition de glyphes, de polices ou de règles orthotypographiques. Les caractères sont par essence des unités abstraites de contenu textuel, mais il a bien fallu en publier *une* représentation pour pouvoir les identifier. Le texte d'Unicode insiste bien sur le fait que ces tableaux ne sont que des tableaux de glyphes *représentatifs* pour ces caractères, glyphes fournis à titre d'*exemples*.

Pour l'écriture latine, cette distinction devrait être manifeste. Il existe de nombreuses belles polices informatisées pour les caractères latins. Le consortium n'a pas conçu ces polices et il n'interdira pas plus à l'avenir aux typographes d'améliorer ou d'étendre ces polices à la lumière de leur conception esthétique et typographique.

Bien sûr l'existence de merveilleuses polices et de maîtres typographes n'empêche pas des amateurs de concevoir des polices laides ! En passant, ce phénomène est antérieur à Unicode. Après tout, c'est le Macintosh qui a lâché ces hordes d'amateurs concevant d'épouvantables polices pour tous les domaines imaginables. Mais dans ce domaine, je serais d'avis de laisser mille fleurs fleurir. Je suis confiant qu'en typographie, contrairement au mauvais argent qui chasse le bon, il est fort probable qu'à la longue la bonne typographie chassera la mauvaise. Après tout, la typographie informatique est encore relativement jeune et les outils continuent de s'améliorer.

PATRICK ANDRIES — Étant donné la taille de l'espace de codage disponible, pourquoi a-t-on unifié les caractères CJC ? Cette unification ne s'est de toute façon pas faite sur une base sémantique, mais souvent simplement en fonction des jeux de caractères où les caractères concernés apparaissaient avant unification – ces points de code représentent-ils alors des « caractères » ? N'aurait-on pas pu affecter une zone à chacune des écritures CJCV ? En quoi l'unification actuelle est-elle meilleure que cette proposition de désunification ?

KEN WHISTLER — C'est ce qu'il fallait faire. On se penche souvent sur ce problème sous un mauvais angle en s'imaginant que les caractères chinois, japonais et coréens sont fondamentalement différents et qu'il a fallu déployer un effort énorme pour les faire entrer dans les mêmes cases et regrouper de force des entités par essence différentes. Pour ceux qui ne n'ont pas une connaissance intime des caractères chinois, cela équivaudrait à dire que les caractères français, allemands et espagnols ne devraient pas être unifiés en un seul jeu de caractères parce que ces langues ont leur propre alphabet et ne partagent pas toutes les mêmes lettres. Mais l'existence et la réussite du Latin-1 comme jeu de caractères qui permet de coder des textes français, allemands et espagnols tout en utilisant les mêmes caractères montrent bien la voie à suivre. (Il est vrai qu'il manquait au latin-1 le caractère œ pour le français.)

Les systèmes d'écritures japonais et coréen (et, également, l'ancien vietnamien) ont tous emprunté des caractères chinois, à l'instar des Allemands qui empruntèrent leurs lettres aux Romains. La différence ne portant que sur le *type* de caractères empruntés (des idéogrammes plutôt que des lettres) et sur le nombre plus important de signes empruntés. Tous en Extrême-Orient comprennent cela – après tout les Japonais ne nomment-ils pas les idéogrammes des « kanji », littéralement des « caractères chinois ». Et bien qu'on ait toujours inventé localement de nouveaux caractères, il demeure que la grande majorité des caractères chinois est commune à plusieurs pays d'Extrême-Orient.

L'astuce pour Unicode consistait à ne pas reproduire fidèlement chacune des normes nationales codant les idéogrammes, car cela aurait produit une multiplication prodigieuse de caractères identiques. Sans cela, plutôt que de coder une seule fois chaque caractère, on aurait abouti à la situation où il aurait fallu mettre en œuvre un mécanisme d'échappement à états : certains numéros de caractère auraient été désignés japonais, alors que les « mêmes » caractères se seraient vus affectés à d'autres numéros pour le chinois. À nouveau, pour reprendre l'analogie du français, de l'allemand et de l'espagnol, cela aurait signifié affecter des numéros de caractères différents à *a*, *b*, *c*

pour chacune de ces langues et à passer d'une zone du codage à l'autre pour pouvoir distinguer les caractères *s* présents dans *six*, *sechs* et *seis.*

Les exceptions à l'unification han mentionnées ci-dessus sont – une nouvelle fois – le résultat de considérations pragmatiques destinées à assurer la compatibilité d'Unicode avec des codages préexistants. Certains jeux de caractères historiques d'Extrême-Orient codaient séparément deux formes légèrement différentes de ce qu'on considère habituellement comme un *même* caractère ou, dans certains cas, attribuaient deux numéros à des caractères parfaitement identiques mais qui se prononcent différemment (ce qui reviendrait à coder deux *c* en français parce qu'on le prononce parfois /s/ et d'autres fois /k/). Ces erreurs dans les jeux de caractères historiques d'Extrême-Orient ont été héritées par Unicode par la « règle de séparation des sources » de l'unification han ou, pour les caractères de compatibilité, par le même processus qui a forcé l'inclusion d'erreurs similaires présents dans d'autres jeux de caractères historiques essentiels.

PATRICK ANDRIES — On dit souvent qu'il s'invente ou se découvre de nouveaux idéogrammes chaque jour ; les utilisateurs CJC devront attendre avant de pouvoir les utiliser. Pourquoi n'existe-t-il pas de notation générative qui permette d'accéder aux idéogrammes et d'éliminer ce délai ?

KEN WHISTLER — Il est exagéré de dire que de nombreux caractères sont sans cesse inventés ou découverts.

Considérons d'abord l'aspect invention. Bien qu'il soit vrai qu'on invente de nouveaux caractères propitiatoires pour représenter des noms de personne dans certaines régions chinoises et qu'on baptise les chevaux de courses de noms fantaisistes à Hong-Kong, ce genre de choses est désormais désapprouvé par les autorités concernées, précisément parce qu'elles posent des problèmes à leurs propres systèmes informatiques. Et cela pour des motifs qui sont étrangers à Unicode – ce problème affecte tout simplement tous les jeux de caractères préexistants d'Extrême-Orient ainsi que les processus informatiques qui en dépendent. La plupart des nouveaux mots chinois sont crées à partir de mots préexistants plutôt qu'en inventant un nouveau caractère. Il existe un certain nombre d'exceptions bien connues, comme la longue liste de nouveaux caractères utilisés dans la classification périodique des éléments, mais celles-ci sont bien connues et pleinement prises en compte.

Considérons maintenant l'aspect découverte. Ce n'est pas comme si un archéologue découvrait toutes les deux ou trois années une nouvelle Troie et y trouvait des milliers de caractères chinois inconnus jusqu'alors. Les Chinois n'ont cessé de répertorier leurs propres caractères et documents depuis des milliers d'années ; ces répertoires se retrouvent de nos jours reproduits dans des encyclopédies et dictionnaires[17] très

17 Soulignons au passage que le plus grand dictionnaire chinois vers une langue européenne est en français : *Le Grand Ricci*. Composés de 7 volumes réunissant près de 9 000 pages. Le Grand Ricci est publié par Desclée de Brouwer ; ISBN : 2-220-04667-2 et se vend, à l'heure de mise sous presse, au prix de 533,58 € chez Amazon.fr.

complets publiés en Chine, au Japon et en Corée. Quand on découvre de « nouveaux » caractères anciens aujourd'hui, il s'agit de cas très particuliers et rares – la plupart du temps des erreurs d'impression ou des formes très anciennes de caractères. L'immense majorité des caractères contemporains font partie depuis belle lurette d'Unicode et l'addition de nombreux caractères dans Unicode 3.1 permet de couvrir désormais la quasi-totalité des caractères chinois utilisés dans les domaines historiques, lexicographiques, religieux et autres spécialités.

Quant aux notations génératives pour caractères chinois, plusieurs de ces mécanismes ont déjà été inventés, mais ils ont tous échoués quand il s'agissait de coder des caractères chinois dans le domaine de la représentation générale de documents.

Le consortium Unicode a décidé d'adopter un mécanisme universel très simple qui permet de *décrire* des caractères qui ne feraient pas encore partie de la norme – ce mécanisme utilise moins d'une douzaine de symboles visibles permettant d'indiquer à l'aide de caractères déjà codés la manière dont les différentes parties du caractère manquant s'agencent afin de suggérer la forme de ce caractère. Ce mécanisme, la description idéographique[18], n'a d'ailleurs pas été inventé par le CTU. Il a été décrit par l'organisme national chinois dans sa norme nationale, connue désormais sous le nom de GB 18030 et a été intégré dans le standard Unicode pour des raisons de compatibilité.

PATRICK ANDRIES — À votre avis, Unicode replacera-t-il les autres jeux de caractères comme le Latin-1 ou le Latin-9 ?

KEN WHISTLER — À terme, oui. Mais la période de transition sera très longue car la mise à jour de logiciels prend beaucoup de temps – plus particulièrement quand ils sont répartis dans un réseau – et la mise à jour des données prend encore plus de temps.

PATRICK ANDRIES — Selon vous, Unicode va-t-il normaliser d'autres aspects relatifs aux écritures ? Lesquels ? Des aspects typographiques ?

KEN WHISTLER — Pas vraiment. Il n'est pas du ressort du consortium Unicode de suggérer des normes orthotypographiques ou la façon dont les polices devraient être conçues. C'est l'affaire d'autres parties concernées, souvent régionales, de décider de ces choses.

Toutefois, comme pour toute autre technique touchant à l'écriture, il est inévitable que les limites ou les potentialités du traitement informatique des documents auront des conséquences inattendues sur l'évolution des systèmes d'écriture. L'élaboration du standard Unicode n'est qu'une petite partie dans cet ensemble bien plus grand : il permet tout simplement de faire coexister des caractères de différentes écritures dans un même document. Mais je pense que la majorité des textes Unicode continueront d'être monolingues puisque les textes multilingues sont l'exception plutôt que la règle.

18 Voir les pages 313-316 dans <http://iquebec.ifrance.com/hapax/pdf/Chapitre-11.pdf>, plus particulièrement les caractères U+2FF0-2FFB.

PATRICK ANDRIES — Avec près de 800 000 points de code disponibles pour le codage de nouveaux caractères, pensez-vous qu'Unicode codera d'autres caractères idéographiques ou iconographiques ?

KEN WHISTLER — Il est sûr qu'Unicode comprendra à terme d'autres caractères idéographiques ; après tout, plusieurs systèmes d'écritures historiques d'Extrême-Orient, comme le tangout[19] et le k'itan[20], ont emprunté le *concept* d'écriture idéographique à la Chine pour ensuite inventer leurs propres idéogrammes plutôt que d'emprunter, comme les Japonais, les idéogrammes chinois.

On peut également être certain que de nombreux jeux de symboles, de différents types, s'ajouteront à Unicode.

Toutefois, il est peu probable qu'on aille jusqu'à ajouter des systèmes de signes généraux comme, par exemple, la signalisation routière, les signes d'interdiction de fumer ou le genre de symboles de danger que l'on retrouve sur le matériel électrique. Ces signes, bien qu'ils aient évidemment un sens, ne font pas partie du domaine des systèmes d'écritures tel qu'on l'entend habituellement. Le standard Unicode se préoccupe de coder les caractères que l'on rencontre dans les textes et non pas de coder tous les signes ayant un sens.

PATRICK ANDRIES — Croyez-vous que l'on enseigne suffisamment l'interna–tionalisation et Unicode à l'heure actuelle dans les universités ?

KEN WHISTLER — À vrai dire, non. L'internationalisation est le parent pauvre dans la plupart des programmes universitaires d'informatique. Ce n'est d'habitude pas un domaine de recherche ou de spécialisation et on ne lui accorde pas l'importance qu'il mérite, c'est également le fait d'autres problèmes pratiques qui intéressent l'industrie informatique, dans les cours d'introduction de conception logicielle ou de génie logiciel.

Malheureusement, les informaticiens fraîchement diplômés ne découvrent la complexité du domaine des jeux de caractères et de l'internationalisation qu'une fois embauchés ! Ce manque de place accordé à l'internationalisation dans les programmes universitaires n'est que le corollaire de la méconnaissance de l'importance de l'internationalisation dans la mise en marché d'un logiciel destiné au marché mondial. Toutefois, développer correctement des logiciels adaptés aux marchés allemands et japonais peut avoir un impact non négligeable sur le bilan d'une entreprise.

PATRICK ANDRIES — Comment voyez-vous l'avenir d'Unicode ? Pensez-vous qu'un jour viendra où le travail sera terminé ?

---

19 Appelée *Hsi-hsia* par les Chinois, le royaume des Tangouts, un peuple de langue tibétaine, fut détruit en 1227 par le mongol Gengis Khan, quelque mois avant sa mort.

20 Les K'i-tan (en transcription chinoise) ou Khitaï (en transcription arabo-persane) appartenaient à la famille mongole. Établis en Chine du Nord depuis le IV^{e} siècle, ils y fondent la dynastie des Liao (947-1125). Leur empire sera anéanti en 1125 par les Jou-tchens (encore appelés Djurtchät) qui fondèrent la dynastie Kin (ou Chin).

KEN WHISTLER — Je pense que cela va encore nous prendre dix ans pour coder les écritures du monde. Il manque encore un certain nombre d'écritures historiques, notamment les hiéroglyphes égyptiens, la plupart des cunéiformes et l'avestique – mais des propositions de codage pour ces écritures sont en bonne voie.

La majorité de l'industrie informatique, cependant, considère ce travail sur des écritures mineures ou historiques comme du ménage de peu d'importance. Les écritures les plus importantes de la planète sont toutes bien traitées par la version actuelle d'Unicode et l'on peut penser qu'Unicode permet déjà la majorité des applications informatiques qui sont de son ressort. C'est pourquoi, certains considèrent que le standard Unicode est, peu ou prou, terminé.

Pour ma part, je suppose qu'on peut dire que je fais partie des rêveurs. Je continue de penser qu'Unicode devrait permettre de représenter les textes du monde entier et servir de la sorte à préserver le patrimoine culturel de l'humanité.

PATRICK ANDRIES — Pensez-vous que les grandes entreprises continueront à soutenir et à adopter Unicode comme elles l'ont fait par le passé alors que des écritures de plus en plus rares se voient codées ? Pourquoi Microsoft ou Sybase devraient-ils se préoccuper du bougui[21] ou du tifinagh ?

KEN WHISTLER — La raison principale pour laquelle Microsoft et Sybase devraient se préoccuper du bougui ou du tifinagh n'est pas liée à la taille du marché logiciel parmi les Bouguis ou les Berbères, mais simplement parce qu'il est de leur intérêt que tous les logiciels au monde n'utilisent qu'*une seule* représentation de texte. Ce que Microsoft veut éviter c'est la multiplication des jeux de caractères, restreints à de petits marchés et comportant des particularités de mise en œuvre. Il est bien plus rentable, pour tout le monde dans l'industrie informatique, de participer à l'élaboration d'un codage universel plutôt que devoir vivre avec sa solution de rechange.

Il faut également imaginer les conséquences légales potentielles. À un moment donné, l'accès à l'informatique deviendra sans doute un genre de droit fondamental. Les minorités linguistiques exigeront, à juste titre, que les ordinateurs prennent en charge leur langue, un peu comme les handicapés demandent à avoir accès à des bâtiments ou aux ordinateurs. Ne pas prendre en compte une telle évolution serait d'une myopie extrême.

PATRICK ANDRIES — Avec le recul, si vous deviez créer un nouveau jeu de caractères universel, que feriez-vous différemment ?

KEN WHISTLER — Voilà une question difficile. L'histoire du Standard Unicode est un faisceau de choix pragmatiques justifiés par la réalité historique. Il est difficile d'envisager des solutions de rechange à certains de ces choix. La plupart des solutions proposées pour rendre Unicode plus cohérent, plus élégant ou plus esthétique se heurtent à un autre principe, de sorte que la réussite même du standard aurait pu être hypothéquée.

21 Écriture des Célèbes (Indonésie).

Mais je regrette la rancune engendrée par la première confrontation des partisans de l'ISO/CEI 10646 dans sa première mouture et les partisans du nouveau modèle qu'était le standard Unicode. Cette confrontation a nourri le mythe tenace d'une opposition entre la 10646 et Unicode, mythe qui a perduré bien après que les deux comités ont appris à s'entendre et ont synchronisé leurs travaux dans l'intérêt de tous. Cette opposition initiale a inutilement désorienté les gens et a, à coup sûr, retardé l'adoption d'Unicode par la communauté Unix et par d'autres normes.

PATRICK ANDRIES — M. Ken Whistler, je vous remercie d'avoir bien voulu répondre à ces questions.