

Université de Bretagne Occidentale

Faculté des lettres et sciences sociales Victor Ségalen

Mastère 2 Rédacteur/Traducteur



Yoann Le Bleis

Indexation et Rédaction techniques



NexWave
Jun - Septembre 2006

Enseignant tuteur
Janet Ormord

Maître de stage
Valentine Ogier-Galand

“Let’s be blunt about this: indexing is a skill, and you won’t ever get good at it if you don’t study the theory, then practice applying that theory until you develop a reasonable level of skill.”

HART Geoff. *Recipe for Designing Usable Documentation.*

Remerciements

Valentine Ogier-Galand et Michel Verceil pour leur temps, leurs précieux conseils et explications.

Toute l'équipe de NexWave Solutions pour leur accueil.

Janet Ormrod pour son suivi au long de ce stage.

Julie et Damien Giraud pour leur hospitalité à Dublin et Montpellier, respectivement, et sans qui, disons-le, j'aurais fini par dormir sous les ponts !

Table des matières

Un stage singulier 4

Une recherche de stage difficile 5

- 1 – Premiers contacts 5
- 2 – Un voyage stérile ? 6
- 3 – Des efforts récompensés 7
- Conclusion 9

NexWave, une entreprise de pointe 10

- 1 – Innovation 10
- 2 – Organisation de l'entreprise 11
- 3 – Le service documentation 12

Indexation : théorie et pratique 15

- 1 – Cadre de travail 15
- 2 – Formations 17
- 3 – Réalisation 21
- 4 – Validation 26
- Conclusion 29

Relectures.30

- 1 – Propositions d'améliorations 30
- 2 – Livret d'accueil 31
- 3 – Documentation Eclipse 31
- 4 – « Good written and spoken knowledge » 32

Une riche expérience, à bien des égards 34

Bibliographie 38

Annexes 39

Introduction

Un stage singulier

Malgré les soins apportés à la conception de mon CV, ma recherche de stage n'a pas été des plus aisées. En effet, de nombreux rebondissements, qu'il me semble intéressant de rapporter dans ce document, dus à des erreurs de ma part mais aussi à quelque manque de chance parfois, ont précédé la réponse positive d'une entreprise qui me proposait une mission des plus singulières en rédaction technique : l'indexation de leur documentation existante.

Il m'a tout d'abord fallu assimiler les principes de la technologie de pointe dans laquelle l'entreprise est engagée. NexWave ne s'adresse pas à des novices en informatique comme pourraient le faire des développeurs de traitement de texte. En effet, les outils mis au point par les ingénieurs de la startup montpelliéraine s'adressent à d'autres ingénieurs. Avant de pouvoir commencer ma mission, il m'a donc fallu étudier certaines bases de la programmation, ainsi que les spécificités de la technologie de NexWave.

Si nous avons étudié durant l'année de Mastère 2 Rédacteur/Traducteur les principes de la rédaction technique, nous n'étions en revanche que très vite passé sur la notion d'index. Il a donc fallu aussi que je m'initie aux enjeux et techniques de l'indexation. Je me suis vite aperçu que le métier d'indexeur nécessite un réel savoir-faire qu'il est parfois difficile d'assimiler par ses propres moyens.

Chapitre 1

Une recherche de stage difficile

Malgré mes premiers CV envoyés dès décembre, de nombreux contacts pris dans des entreprises de natures parfois très différentes, et un séjour à Dublin où j'ai pensé pouvoir trouver de bonnes opportunités, ce n'est qu'à la mi-mai que j'ai reçu une réponse positive de la part de NexWave Solutions, à Montpellier.

1 – Premiers contacts

J'ai commencé assez tôt à me pencher sur ma recherche de stage, en rédigeant mon CV durant la première quinzaine du mois de décembre. En mettant à plat mes acquis j'ai constaté que j'avais un profil qui pouvait correspondre à certaines entreprises : un bac scientifique, de bonnes bases en informatique, des connaissances dans des domaines assez variés, et une spécialisation dans un domaine qui peut regrouper plusieurs de ces compétences. J'ai donc choisi pour mon CV de mettre en exergue mon profil pluridisciplinaire (cf. [annexe 1](#)).

Étant donné que j'avais déjà effectué un stage auprès d'un traducteur juridique en Espagne à la fin de mon année de licence LEA, je souhaitais pour le Mastère m'essayer à un autre domaine, et ce dans un pays anglophone afin d'améliorer mon anglais encore, d'après moi, trop faible pour le domaine dans lequel je me destine à travailler.

Parce que Dublin est une ville où se sont installées de nombreuses multinationales travaillant dans les nouvelles technologies, mais aussi parce que j'avais des amis là-bas et une certaine attirance pour cette ville, j'ai orienté mes recherches vers des entreprises de la capitale irlandaise.

J'ai donc envoyé ma candidature spontanée pour un poste de stagiaire en traduction dans plusieurs entreprises d'informatique, mais aussi de mécanique – domaine qui me plaisait beaucoup au lycée –, sans qu'aucune ne m'envoie de réponse.

J'ai par la suite répondu à une annonce de choix : un poste de traducteur chez Microsoft sur lequel j'ai longtemps compté. Non content d'avoir une expérience avec le géant de l'informatique, il s'agissait de travailler à Dublin ! Malheureusement le test que j'ai passé pour ce poste n'a pas suffi à le convaincre, et ils m'ont gentiment remercié. Selon moi, c'est mon anglais fébrile lors des contacts téléphoniques qui a fait mauvaise impression. Une conclusion très décevante qui confirmait mon besoin de séjourner dans un pays anglophone.

J'ai par ailleurs retenu l'attention d'une entreprise de localisation de jeux vidéo : Keywords International. Elle était à la recherche de testeurs, et mon profil axé sur les langues les intéressait pour la validation des textes en français. J'ai profité de vacances à Dublin pour rencontrer le responsable du département de test à qui j'ai fait bonne impression et qui m'a promis un poste dès la fin des cours.

Au début du mois d'avril, j'ai donc naturellement volé vers Dublin où j'avais un poste assuré, et sans doute d'autres opportunités dans les nombreuses entreprises de traduction ou d'informatique.

2 – Un voyage stérile ?

Dès mon arrivée à Dublin, j'ai contacté Keywords afin de convenir d'une date pour commencer mon stage. Malheureusement les projets sur lesquels ils avaient prévu de me faire travailler avaient pris du retard et ils ne pouvaient me prendre en tant que testeur qu'à partir de la fin juin. Je leur ai donc

proposé de commencer mon stage en tant que traducteur, puis passer au test au moment venu. Ils ont accepté, malgré leur habitude de faire commencer les nouveaux arrivants dans le département de test/relecture, et à la condition que je réussisse leur test de traduction. Il était constitué de trois parties : un document sur les logiciels-espions, la présentation d'un jeu vidéo, et des « trucs et astuces » pour la console X-Box. Malheureusement j'ai commis de grossières erreurs que je peux mettre sur le compte du manque de concentration dû aux conditions inadaptées dans lesquelles je l'ai passé, et sans doute de la fautive légèreté avec laquelle j'ai pris cet « examen ». Il n'était donc plus question de travailler chez Keywords, et je me retrouvais sans aucune alternative.

J'ai par la suite contacté toutes les agences de traduction de Dublin, qui refusaient parce qu'elles travaillaient avec des indépendants, ou qui n'avaient tout simplement pas de postes ouverts.

Assez rapidement je me suis tourné vers les sites de recherche d'emploi de type Monster afin de trouver des entreprises demandeuses de services de traduction ou de rédaction technique. Même si mon expérience ne me permettait pas de postuler, j'espérais que certaines entreprises trouvent un intérêt à prendre de la main-d'œuvre peu onéreuse, toujours sans succès.

3 – Des efforts récompensés

Puisque mes efforts semblaient vains, que l'Irlande semblait ne pas avoir de stage pour moi, et que le temps jouait en ma défaveur, je me suis résigné à

Senior Technical Writer
 Our client has a vacancy for a contract role for an experienced Technical Writer. The Technical Writer position in XSIL gives exposure to creating both hardware and software-related ...[\[more\]](#)

Relevant Work Experience: 5 to 7 Years
Career Level: Experienced (Non-Manager)
Education Level: Bachelor's Degree
Job Type: Temporary/Contract/Project
Job Status: Full Time

TECHNICAL WRITER
 Technical Writer Overview Wonderful opportunity with leading Financial Software House based in Dublin. 3+ years experience in technical writing or a similar role required. Superb Prospects. ...[\[more\]](#)

Relevant Work Experience: 2 to 5 Years
Job Type: Permanent
Job Status: Full Time
Salary: From 35,000.00 to 40,000.00 EUR per year

Figure 1. extrait des offres d'emploi du site [Monster.ie](#)

répondre aux annonces de Kap'stage, ainsi qu'à celles qui avaient été adressées au Mastère.

J'ai ainsi retenu l'attention de plusieurs entreprises, notamment de Legal Suite (Paris) qui m'a fait passer un entretien par vidéoconférence. Malheureusement j'étais en concurrence avec un rédacteur technique plus expérimenté. J'ai également passé un test de traduction pour un poste dans la communication interne d'une grande entreprise d'aéronautique, mais entre temps j'avais obtenu des contacts avec des entreprises qui m'attiraient bien plus.

SDL, leader mondial dans le marché de la traduction, m'a fait passer un test pour un poste de traducteur débutant dans ses bureaux de Sheffield. Test constitué de trois parties (marketing, informatique et questions sur mes connaissances techniques) et que j'ai passé avec succès – enfin une certaine reconnaissance ! Cependant il ne représentait que la première étape d'un recrutement beaucoup plus long et le temps venant à manquer, j'ai donc dû leur faire part de mon impossibilité de continuer. Cela étant, ce test réussi n'est pas perdu et il n'est pas exclu que j'ai l'occasion un jour de « relancer » SDL sur la base de cette candidature avortée.

L'offre de stage de NexWave (cf. [annexe 2](#)), reçue par le biais de la liste courriel du Mastère, s'avérait elle aussi très intéressante. Après un test satisfaisant (cf. [annexe 3](#)) et un entretien téléphonique plutôt informel avec la responsable de la documentation technique, j'ai décidé de mettre fin à ma recherche ainsi qu'à mon séjour en pays anglophone en achetant un billet pour Montpellier.

Conclusion

Si la plupart de mes démarches pour trouver un stage se sont avérées infructueuses, j'ai néanmoins de bonnes leçons à tirer concernant la recherche d'emploi, la préparation d'entretiens et le passage de tests de recrutement.

Par ailleurs, mon séjour en Irlande qui pourrait passer pour des vacances aux yeux de certains est loin d'avoir toujours été de tout repos, et il m'a permis de prendre un peu de confiance en mon niveau d'anglais, ainsi que de faire de riches rencontres et de visiter un pays magnifique !

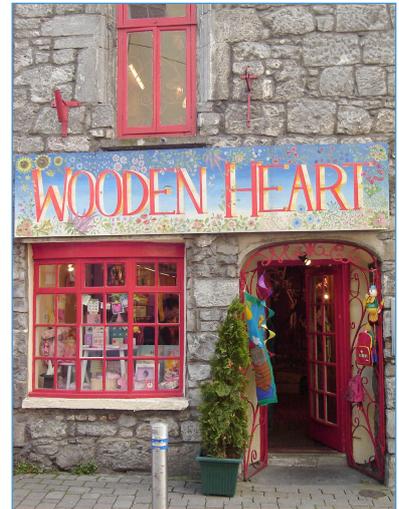


Figure 2. une boutique du centre de Galway

Chapitre 2

NexWave, une entreprise de pointe

À mon arrivée chez NexWave Solutions, il me faut m'habituer à une entreprise bien différente de celles que j'ai pu voir jusqu'à présent : il y est question de l'informatique embarquée, domaine d'expertise dont j'ignorais l'existence auparavant, on y tient des vidéoconférences avec des ingénieurs en déplacement au Japon ou en Corée, et on y maîtrise parfaitement un réseau interne très élaboré.

1 – Innovation

Les entrepreneurs à l'origine de la création de NexWave avaient pour projet en 1998 de répondre à un besoin des fabricants d'appareils électroniques grand public (téléviseurs, lecteurs DVD, téléphones portables, etc.) de « recycler » les programmes qu'ils créaient pour une plate-forme précise afin de pouvoir les réutiliser sur d'autres dispositifs, et ainsi réduire leurs coûts de développement logiciels et leur temps de mise sur le marché de leurs produits.

Aujourd'hui elle propose aux grands noms du secteur une technologie novatrice et des outils permettant de créer et d'assembler

des « composants » correspondant à des sous-programmes indépendants et portables, c'est-à-dire pouvant être embarqués dans des dispositifs électroniques de natures très différentes. Un même composant permettant de

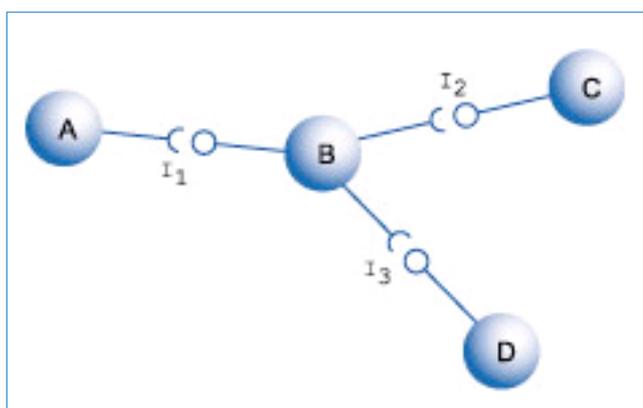


Figure 3. modélisation de la technologie de NexWave : le composant B exporte des services à A via l'interface I1

décoder une image GIF, par exemple, pourra être embarqué dans un appareil photo et un téléphone portable sans modifier son code-source.

NexWave couvre toute la chaîne de production d'un logiciel à base de composants. Des plugins pour l'environnement de développement libre Eclipse facilitent le codage des composants. NexDev et NexBuilder permettent de les compiler puis de les assembler. Enfin NexGUI et NexGUIBuilder, produits phares de NexWave, sont dédiés à la création d'applications graphiques, ou *widgets*, permettant à un utilisateur d'interagir avec son appareil électronique. Avec ces derniers logiciels, NexWave facilite par ailleurs la localisation des *widgets* créés aux différents marchés de ses clients.

De plus, NexWave se veut entreprise-conseil en aidant ses clients à se réorganiser pour adopter sa technologie innovante. Grâce à elle, il est en effet possible de séparer complètement la phase de création des composants et celle de l'assemblage, qui peut être effectué par des personnes ayant très peu de connaissances en programmation. NexWave souhaite enfin favoriser les échanges de composants entre les acteurs du secteur de l'informatique embarquée.

2 – Organisation de l'entreprise

NexWave, startup constituée de près de 30 salariés, conduit le cœur de ses activités à Montpellier, mais est également représentée en Californie, au Japon, en Corée et en Chine par différents mandataires ingénieurs et commerciaux. De par sa vocation internationale et la présence de non-francophones dans l'entreprise, l'anglais est couramment utilisé, notamment à l'écrit ou lors de réunions.

La plupart des salariés de NexWave travaillent dans un *openspace* où différents îlots regroupent les équipes de programmeurs par spécialités (cf. orga-

nigramme en [annexe 4](#)). Chaque programmeur peut par ailleurs être associé à une équipe-projet afin de répondre à une commande spécifique d'un client.

L'organisation du travail repose enfin sur un réseau interne très élaboré qui permet le regroupement, le partage et l'archivage des informations et des fichiers afin de faciliter le travail en groupe, ainsi que la gestion très efficace des bogues découverts dans les programmes ou dans la documentation.

3 – Le service documentation

En étroite collaboration avec les ingénieurs des différentes équipes, les rédacteurs techniques sont chargés de la production et de la maintenance des documents de référence destinés aux utilisateurs de la technologie à base de composants de NexWave, ainsi que des manuels d'utilisation des différents outils qu'elle propose. Par ailleurs, l'équipe documentation s'occupe de la formation des nouveaux arrivants en interne et des ingénieurs des entreprises clientes en externe.

Si sa charge de travail de travail est considérable, elle n'est pourtant constituée que d'une rédactrice technique à plein temps et d'un ingénieur qui a récemment choisi de la rejoindre à mi-temps. Il est par ailleurs intéressant de noter qu'un poste vacant de rédacteur confirmé ne trouve pas preneur, faute de candidats au profil vraiment adapté au domaine très pointu et à l'audience constituée d'ingénieurs : rares sont les techniciens qui passent à la documentation, et rares sont les rédacteurs qui ont les connaissances techniques suffisantes au poste proposé.

Chacun des deux rédacteurs actuellement présents a bénéficié d'une formation scientifique et d'une très large expérience dans la programmation. Malgré mon bac scientifique, mon parcours suivi en faculté des lettres et sciences sociales me démarque très nettement du « profil-type ». Je suis en effet le seul

dans l'équipe documentation, et plus largement dans l'équipe production, à ne pas avoir de diplôme du supérieur dans les sciences ou l'informatique. Malgré cela, il m'a fallu comprendre les principes de la technologie de NexWave, ce qui impliquait d'acquérir quelques bases en programmation et une certaine culture informatique dépassant notablement ce que nous avions pu voir dans les cours dispensés à Brest.

Outre les programmes de NexWave et les logiciels habituels de bureautique de Microsoft (Word, Excel, PowerPoint et Outlook), l'équipe documentation est dotée de logiciels plus spécifiques à leurs activités. On peut compter parmi eux SnagIt pour les captures d'écran très utiles en rédaction technique, mais surtout Adobe Distiller pour créer des documents au format PDF, Adobe Illustrator, logiciel de dessin vectoriel, et FrameMaker. Ce dernier, souvent mentionné dans les offres d'emploi en rédaction technique, m'était totalement inconnu avant mon arrivée à NexWave. C'est pourtant sur celui-ci que j'étais amené à réaliser l'index. Sans doute est-ce grâce à mes connaissances d'autres logiciels distribués par Adobe (InDesign, PhotoShop), que j'ai rapidement assimilé les fonctions qui me concernaient.

Chaque mercredi, une réunion de l'équipe documentation était organisée afin de faire un bilan des tâches réalisées au cours de la semaine par chacun. C'était une occasion de faire part de mes difficultés et de tenter de les résoudre tous ensemble. Puis de nouveaux objectifs étaient fixés pour la semaine suivante ou pour une date plus lointaine selon l'importance de la tâche. Le compte-rendu de ces réunions était transmis au supérieur hiérarchique, à savoir le Responsable Développement.

Il est également intéressant de signaler le passage progressif au *single sourcing* dans la documentation technique. Cette documentation à source unique, basée sur le système Darwin information typing architecture (DITA), permet de créer, à partir des mêmes textes, de la documentation sous formes diver-

ses (PDF, aides en ligne, etc.) grâce la séparation de la forme et du fond que permet le langage XML. Ceci dit, l'intégration du système DITA n'étant pas assez avancée à mon arrivée, il n'a eu aucune incidence sur mon travail d'indexation.

Chapitre 3

Indexation : théorie et pratique

«Rattaché à la rédactrice technique, vous assurerez l'indexation de la documentation technique existante : identification des concepts clés, hiérarchisation, marquage et génération des tables d'index.»

C'est en ces termes qu'était définie ma mission dans l'offre de stage. Pour l'accomplir, plusieurs étapes ont été nécessaires. Nous nous intéresserons à l'indexation d'un seul des manuels de la documentation de NexWave, le *Programmer's Manual* (cf. [annexe 6](#)).

1 – Cadre de travail

1.1 cahier des charges

Il s'agissait de concevoir et réaliser *ex-nihilo* un index, ce qui impliquait de faire moi-même des choix, que ce soit dans la mise en forme ou dans la structure et la répartition des termes dans l'index. Pour assurer la pérennité de l'index créé, il était par conséquent primordial de documenter ces choix, de les expliquer et de les illustrer afin que les personnes qui travailleraient à sa mise à jour par la suite puissent aisément maintenir la cohérence, que ce soit sur le fond ou sur la forme.

L'index à réaliser était de type *embarqué*, c'est-à-dire qu'il faisait partie intégrante du document indexé et était généré automatiquement à partir des *marqueurs* insérés manuellement dans le texte. Il avait pour principal avantage la mise à jour simplifiée du document : il suffisait en effet de régénérer l'index pour actualiser les numéros de pages, par exemple. Cependant, nous verrons par la suite que ce type d'index comportait certains inconvénients.

Le premier document à indexer, le *Programmer's Manual* (cf. [annexe 6](#)), est un manuel destiné à définir des concepts et aider un programmeur à concevoir et écrire un logiciel à base de la technologie composants de NexWave. Ce n'est pas un manuel de références à proprement parler, dans le sens où il peut être lu de façon linéaire, mais peut néanmoins être consulté comme tel. Si l'on peut se poser la question de savoir si un index est utile dans un manuel de références classées par ordre alphabétique, la nécessité d'en avoir un dans le *Programmer's Manual* est indiscutable au regard de sa taille (près de 200 pages) et de son utilisation présumée : lecture ponctuelle de définitions de concepts ou de règles et conseils pour la programmation en suivant le modèle de NexWave.

Le public est composé d'ingénieurs en informatique, donc très familier des langages de programmation et du vocabulaire technique du domaine. Ils sont, pour la plupart, non-anglophones natifs. Certains ont suivi un ou plusieurs stages de formation avec les équipes de NexWave, mais ceux qui se mettent à la technologie NexWave sans assistance extérieure doivent aussi pouvoir le consulter et utiliser son index. Il convenait donc d'utiliser le vocabulaire commun à l'ensemble de la communauté des ingénieurs en informatique, mais aussi certains termes qui se font concurrence, et de rediriger les utilisateurs vers les termes utilisés dans le cadre de la technologie de NexWave.

1.2 contexte et outils

Les documents de travail étaient partagés par l'ensemble des membres de l'équipe documentation, et je ne pouvais donc pas en permanence travailler sur la version électronique du manuel. Lorsqu'elle était utilisée par les autres rédacteurs techniques pour ajout de sections ou pour révision, je m'attélais à relire l'index sur papier, ou à étudier sa structure et le choix des entrées. Ce partage des documents de travail se faisait par le biais du réseau interne. Une

fois les modifications apportées par un membre de l'équipe, le document était publié sur un répertoire commun accompagné d'un fichier dans lequel étaient rapportées les modifications internes et externes (visibles par le client ou non), et son numéro de version était incrémenté.

L'index devait être réalisé à l'aide du logiciel Adobe FrameMaker 6.0 très utilisé dans la production de documents techniques. C'est un outil puissant qui permet la création et la mise en forme en WYSIWYG (what you see is what you get) de gros documents à l'aide de maquettes et de pages de références, ainsi que l'export des livres créés en PDF, XML ou HTML. L'[annexe 5](#) est une capture d'écran de sa fenêtre principale.

2 – Formations

Avant de commencer l'indexation à proprement parler, il m'a fallu apprendre les principes de deux domaines complètement nouveaux pour moi, à savoir la technologie de NexWave et l'indexation. Cette formation théorique à ces deux domaines d'expertise, qui n'a pas été exempte de difficultés, a occupé près de l'intégralité du premier mois de stage.

2.1 technologie NexWave

Si mon apprentissage de la technologie de NexWave n'avait évidemment pas pour but de faire de moi un développeur de logiciels, il m'a néanmoins fallu, au delà du cadre strict de la spécialité de la startup, assimiler certaines bases et le vocabulaire courant de la programmation. En effet, pour pouvoir rendre un index utile à un programmeur, il me fallait savoir faire la différence entre ce qui était propre à NexWave et ce qui pouvait se retrouver dans d'autres contextes ou technologies.

Cette formation très technique, qui m'a, de manière anecdotique, rappelé les programmes que j'écrivais au lycée pour mes cours de mathématiques ou

pour jouer au Black-Jack sur ma calculatrice graphique, s'est faite par plusieurs moyens.

La majeure partie de cet apprentissage s'est faite par la lecture du *Programmer's Manual* destiné à décrire la technologie à base de composants de NexWave et à guider les programmeurs dans le développement de logiciels selon ce modèle. Si mon avancée dans cette tâche a été très lente à mon arrivée, j'ai pu, à mesure que j'intégrais les principes fondamentaux de la programmation et de la technologie, accélérer le rythme et me familiariser avec le style et le vocabulaire, puisque tel était aussi le but de cette lecture.

Pour m'aider dans mon avancée, j'ai pu compter sur deux sources d'informations. D'une part, mes collaborateurs étaient toujours enclins à me donner des explications très éclairées sur tels ou tels concepts propres à NexWave ou à la programmation. D'autre part, j'ai eu très souvent recours à Internet pour trouver des définitions de termes techniques, que ce soit sur Wikipedia, sur des dictionnaires en ligne, ou sur des sites proposant des cours de programmation (cf. [bibliographie](#)).

Enfin, j'ai suivi une formation de trois jours destinée aux ingénieurs intégrant l'équipe de NexWave et leur permettant d'acquérir les bases de la technologie et des outils développés par l'entreprise. Aux côtés de deux nouveaux arrivants, j'ai pu me familiariser au discours tenu face à des techniciens, et avoir un aperçu de leur point de vue. Si j'ai pu suivre confortablement une grande partie des aspects théoriques de la formation pour avoir au préalable étudié le *Programmer's Manual*, je dois avouer que je n'aurais jamais pu réaliser les exercices pratiques sans l'aide des formateurs.

À la suite de ces trois jours, j'ai néanmoins pu réaliser par moi-même les tutoriaux les plus simples, et me concentrer le sujet de mon stage : l'indexation.

2.2 indexation

« L'index est aussi important pour une documentation qu'une documentation l'est pour un produit. »¹ Cette réflexion est récurrente dans les documents sur l'indexation technique que j'ai pu trouver dans mes recherches sur internet (cf. [bibliographie](#)) pour comprendre les enjeux et apprendre les techniques qui aller me permettre de mener à bien ma mission.

En effet on peut se demander à quoi sert d'avoir une documentation bien conçue, bien écrite, si l'utilisateur n'arrive pas à y trouver l'information dont il a besoin. L'index est un outil fondamental dans cette recherche, et il est sans doute davantage consulté que la table des matières, d'où la nécessité d'apporter le plus grand soin à sa conception.

L'Académie Française définit l'index comme étant une « *table alphabétique des sujets traités ou des mots cités dans un livre, un document, un ensemble de documents, donnant les références des pages où ils apparaissent et, parfois, la définition des termes, ou diverses indications.* »

Nancy C. Mulvany, dans son ouvrage intitulé *Indexing Books*, insiste sur le caractère écrit de l'index par opposition à la génération automatique d'une concordance qui consiste en une liste des mots apparaissant dans un document et dans laquelle manquent l'esprit d'analyse et de synthèse dont un indexeur doit faire preuve pour réaliser sa tâche.

Voici les points-clés de la démarche qui doit être adoptée pour aboutir à un bon index :

- ▶ Distinguer les occurrences pertinentes d'un concept et les simples mentions sans apport d'informations sur le sujet.

¹ Lori Lathrop (ancienne présidente de l'American Society of Indexers) dans une interview au webzine Inkspot

- ▶ Analyser le texte, le lire intelligemment afin de d'identifier les concepts qui sont traités dans le texte, et ce même s'ils ne sont pas mentionnés textuellement (références implicites).
- ▶ Synthétiser l'information, c'est-à-dire formuler le texte des entrées de l'index de manière claire et concise.
- ▶ Reprendre la terminologie utilisée dans le texte, et réorienter l'utilisateur qui utiliserait des termes différents vers ceux de l'auteur du document à l'aide des références croisées.
- ▶ Regrouper les informations sur un même sujet qui sont disséminées dans le texte.
- ▶ Structurer l'information, c'est-à-dire respecter la hiérarchie entre les concepts grâce à l'utilisation appropriée du système des entrées et sous-entrées, et créer des liens entre les concepts.
- ▶ Agencer les entrées, et mettre l'index en forme de manière appropriée.

Comme en rédaction technique, un index doit être conçu en gardant toujours en tête son utilisateur final et ses différents contextes d'utilisation. Il doit être adapté à un public le plus large possible, quel que soit son niveau, débutant ou expérimenté.

Fort de connaissances et d'une certaine culture acquises en fréquentant les sites de la communauté des rédacteurs techniques et des indexeurs, je pouvais alors entrer dans le vif du sujet.

3 – Réalisation

3.1 choix des entrées et structure de l'index

Le choix des entrées et l'établissement de la structure de l'index sont les étapes qui m'ont posé le plus de problèmes. C'est là que des choix stratégiques doivent être effectués et que les qualités de visionnaire d'un indexeur sont primordiales. Il est en effet nécessaire de prévoir la manière selon laquelle l'utilisateur d'un index va chercher les termes dont il a besoin et quelles sont les entrées qu'il va sélectionner pour trouver telle ou telle information.

Comme le rappellent souvent les indexeurs professionnels dans leurs publications sur le sujet, le choix des entrées, le recul par rapport au texte nécessaire à la distinction de la simple évocation d'un concept et d'une vraie discussion sur le sujet, ou encore le rapprochement d'une information et d'un concept qui n'est pas mentionné explicitement, sont avant tout une affaire d'expérience. Beaucoup de temps – peut-être trop – et d'essais m'ont donc été nécessaires à l'indexation et à la structuration de l'information du *Programmer's Manual*. En effet, il n'existe pas à ma connaissance de méthodes clés en main pour être efficace et performant dans le choix des entrées. Tout dépend du cas, pourrait-on dire ! Certaines notions très importantes dans la programmation m'ont particulièrement posé problème : il m'a par exemple été difficile de choisir les informations à mettre sous l'entrée *Makefiles*, par exemple, qui désigne un fichier dans lequel on décrit des règles de compilation. En effet de très nombreuses informations peuvent le concerner à première vue, mais seule une poignée de sous-entrées a subsisté à mes multiples révisions.

Dans un premier temps, j'ai simplement lu le manuel à indexer, et inséré au fur et à mesure des marqueurs aux endroits qui me semblaient appropriés. Au bout d'un chapitre, je me suis aperçu que l'index généré ne respectait pas

la hiérarchisation de l'information, qu'il mettait au même niveau des informations d'importance très inégales, qu'il n'était pas structuré, en somme.

Il me fallait donc trouver un moyen de prendre du recul par rapport à l'organisation de l'information dans le manuel afin de cerner la structure de l'index. On m'a alors suggéré de concevoir un arbre lexical permettant la visualisation de l'organisation et la hiérarchie de la terminologie utilisée. Cependant un arbre, avec sa racine, ses branches, ses nœuds et ses feuilles, ne pouvait servir de modèle à la terminologie de NexWave : quelle est la racine ? Quels sont les parents de ce terme ? Aux liens de parenté verticaux devaient s'ajouter de nombreux liens horizontaux qui compliquaient très vite la lecture de l'arbre, et le rendaient inutilisable pour saisir l'organisation lexicale du manuel et m'aider à structurer l'index.

Le modèle que j'ai adopté pour m'aider à agencer l'index est inspiré de l'arbre lexical, mais s'il ne permet pas de regrouper tous les termes et leurs liens dans un même graphique, il permet de visualiser des conglomerats de termes regroupés autour d'un concept, et ainsi, en se mettant à la place de l'utilisateur de l'index, de prévoir quelles entrées il est susceptible de choisir pour arriver à l'information dont il a besoin. Cette modélisation par « buissons lexicaux », comme il me plaît de l'appeler, rejoint cette citation de Lori Lathrop, ancienne présidente de l'ASI (American Society of Indexers) : « *Un artiste compose d'abord des fragments de son œuvre ; puis s'éloigne pour avoir une vue d'ensemble ; et enfin s'attache à rassembler le tout. Il en va de même pour l'indexation.* »

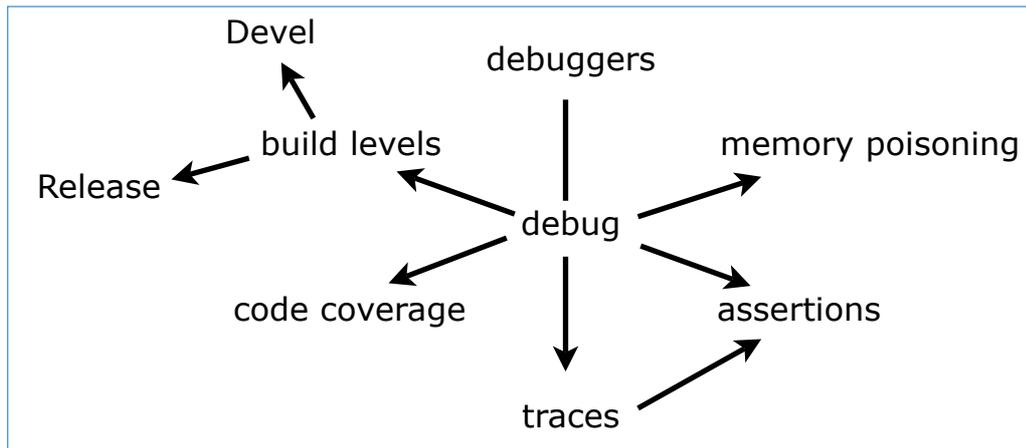


Figure 4. exemple de « buisson lexical »

debug, notion importante dans la programmation, constituera un point d'entrée pour de nombreux autres concepts – pointés par les flèches dans le buisson lexical, et que l'on retrouvera en sous-entrées. On peut en revanche imaginer qu'un utilisateur ira directement à debuggers, sans passer par debug, d'où l'absence de direction dans le segment qui lie les deux termes. L'utilisateur pourra passer par traces pour accéder à assertions, mais aussi directement par debug. Il est à noter enfin que chacun des termes de ce buisson apparaît en entrée principale dans l'index, avec ou sans sous-entrées, cela afin de répondre au degré de connaissance du vocabulaire des différents utilisateurs.

Ces buissons lexicaux m'ont surtout été utiles au début de la création de l'index pour établir la hiérarchisation des concepts-clés. Je m'en suis cependant peu à peu détaché à mesure que j'avais et que la structure de l'index se stabilisait.

3.2 édition de l'index

Outre les contraintes dues au partage des documents de travail par l'ensemble de l'équipe documentation que nous avons traité précédemment, il a fallu que je m'adapte au logiciel FrameMaker et à ses imperfections.

L'intégration de l'index à la documentation nécessitait, en plus de l'adoption de la terminologie utilisée dans le *Programmer's Manual*, sa conformité au style adopté par NexWave : taille des marges, titrages, polices de caractères, en-têtes et pieds de pages, ou encore couleurs, devaient concorder. J'ai donc simplement importé les maquettes établies pour les annexes du *Programmer's Manual* : les marges, plus petites, permettaient la division de la zone de texte en deux colonnes, et l'index étant placé en fin de document, il devait s'inscrire dans la continuité du style des annexes (cf. index en [annexe 8](#)).

Puis il fallait donner des instructions à FrameMaker pour que l'index qu'il générerait automatiquement prenne la forme voulue. Il suffisait pour ce faire d'appliquer les couleurs, les polices et les tailles de caractères aux codes correspondant à l'information à mettre en forme dans les pages de référence de l'index (cf. Figure 4 ci-contre).

```

Symbols[ ];Num-
erics[0];A;B;C;D;E;F;G;H;I;J;K;L;M;N;O
<$symbols><$numerics><$alphabetics>
Niveau1IDM
  Niveau2IDM
  1, 2-3
<$pagenum>

```

Figure 5. codes permettant la génération et la mise en forme automatiques de l'index

L'application de formats de caractères devait également se faire

dans le texte des marqueurs par l'ajout de balises spécifiques (par exemple `<Code>` pour appliquer le style de caractère *Code* jusqu'à la balise de fin `</>`). En effet, toutes les modifications effectuées manuellement dans l'index – ita-lisation d'un *See Also* ou ajout d'un saut de page, par exemple – disparaissaient à la régénération. Idem pour les corrections typographiques, qui n'étaient pas répercutées sur le texte des marqueurs. Je dirais que cette sépa-

ration du texte des marqueurs et du texte généré constitue le plus gros défaut de FrameMaker pour l'indexation. Il est à noter néanmoins qu'il existe des *plugins* pour le logiciel améliorant l'interface pour la gestion des marqueurs, ou permettant de sauvegarder certaines modifications manuelles au sein de l'index.

Le placement des marqueurs dans le texte m'a lui aussi posé problème : pour chaque entrée dans l'index, un marqueur doit lui correspondre. Or plusieurs entrées pouvaient conduire à une même information. Naviguer entre les marqueurs n'étant pas chose aisée dans FrameMaker (la possibilité d'y appliquer des couleurs, par exemple, en aurait facilité la manipulation), il m'a fallu éviter qu'ils ne se superposent en les plaçant de façon parfois un peu hasardeuse autour des mots à indexer.

L'ajout de références croisées (*See et See also*) au sein de l'index a lui aussi dû passer par l'ajout de marqueurs dans le texte-même, alors que ces renvois auraient dû être indépendants du texte source. Comme dit précédemment, tous les ajouts manuels de texte dans l'index disparaissent ou sont déplacés à sa régénération. À moins d'être sûr d'avoir une version définitive de l'index, il était donc déconseiller d'ajouter de telles références à la main. Sur la liste de discussion *TECHWR-L*, on m'a donné une solution à ce problème : pour qu'elles soient générées automatiquement, les références croisées doivent prendre source dans des marqueurs placés dans le texte, comme toute autre entrée, en précisant par une balise l'absence du numéro de page (balise `<$nopage>`). J'ai choisi pour ces marqueurs un peu spéciaux, qui auraient pu se trouver n'importe où dans le livre, de les placer à l'endroit où se trouvait l'information qu'ils pointaient indirectement.

4 – Validation

Deux étapes sont nécessaires à la validation d'un index. D'une part il faut en vérifier la forme, c'est-à-dire la consistance stylistique. Comme pour la relecture de n'importe quel type de documents, j'ai donc vérifié que les mots étaient écrits de la même manière dans tout l'index et que la mise en forme était constante. D'autre part il faut en vérifier le fond. J'entends par là qu'il faut s'assurer que structure de l'index soit cohérente, que les textes des entrées soient clairs et que les informations soient placées aux endroits où les utilisateurs iront chercher l'information dont ils ont besoin.

J'ai dans un premier temps fait relire mon index par des ingénieurs de Nex-Wave. Ils vérifiaient que les concepts qui leur venaient à l'esprit étaient aisément accessibles dans l'index, et que la mise en forme était satisfaisante. Cette méthode, qui aboutissait à des critiques positives, ne me permettait cependant pas de détecter les faiblesses de mon index. Il allait donc falloir que j'aie recours à des tests d'utilisabilité plus formels qui s'intéresseraient plus au fond qu'à la forme de l'index.

4.1 tests d'utilisabilité : théorie

Dans un premier temps, il a fallu que j'étudie ce concept d'utilisabilité actuellement très à la mode dans la communication technique, et que j'apprenne en quoi consistaient les tests qui pouvaient l'évaluer.

La norme ISO 9241-11 donne les lignes directrices de l'utilisabilité en disant qu' « *un système est utilisable lorsqu'il permet à l'utilisateur de réaliser sa tâche avec efficacité, efficience et satisfaction dans le contexte d'utilisation spécifié* ».

Les tests d'utilisabilité doivent donc couvrir les trois domaines suivants :

| Domaine à couvrir | Informations à obtenir | Données à recueillir |
|-------------------|--|--|
| efficacité | Vérifier que les tâches données à l'utilisateur soient réalisées (pas de notion de temps). | <ul style="list-style-type: none"> ▶ nombre de réponses correctes ▶ ratio succès-erreurs |
| efficience | Mesurer les ressources nécessaires pour réaliser ces tâches et la facilité d'apprentissage de l'interface (inclus le temps). | <ul style="list-style-type: none"> ▶ temps de réalisation d'un test ▶ nombre de demandes d'aide |
| satisfaction | Déterminer si le système est agréable à utiliser | <ul style="list-style-type: none"> ▶ expressions de frustration ou de satisfaction ▶ attitude après avoir fini le questionnaire (négative, positive, neutre) |

4.2 questionnaires

Ne pouvant bénéficier des ressources nécessaires à la conduite d'un test très complet, je me suis contenté d'élaborer quelques questionnaires (cf. [annexe 7](#)) et de les soumettre de manière plutôt informelle à des ingénieurs de différentes équipes de NexWave qui, selon leurs missions, n'avaient pas un niveau uniforme de connaissances de la technologie décrite dans le *Programmer's Manual*.

Concrètement, j'interrompais les ingénieurs pendant leur travail et je m'installais auprès d'eux pour leur soumettre les questionnaires. Je les invitais à penser à voix haute afin de noter les termes auxquels ils pensaient après la formulation de la question, et j'observais enfin leur manière d'utiliser l'index : recherche alphabétique pour l'entrée principale, et "survol" des sous-entrées.

J'ai rencontré plusieurs difficultés lors de cette phase de test de l'index.

D'abord j'avais du mal à imaginer des questions. En effet, je me sentais trop influencé par mon index : je connaissais les entrées conduisant aux informations diverses, et il était donc difficile de déterminer celles qui n'étaient pas appropriées et qu'il fallait tester. De plus, je ne me sentais pas toujours en mesure de déterminer quels pouvaient être les problèmes rencontrés par des programmeurs.

Par ailleurs la question de la validité des résultats se posait : j'avais l'impression qu'ils étaient biaisés par le questionnaire lui-même. En effet, les ingénieurs interrogés cherchaient dans l'index les mots-clés qu'ils relevaient dans la question elle-même, alors la phrase était formulée justement de manière à ne pas leur donner les termes indexés. Cela dit, il était parfois inévitable d'éviter les mots-clés, et j'essayais de détourner ce problème en multipliant les termes techniques dans la phrase.

Je pense que des résultats plus justes auraient pu être obtenus en mettant des utilisateurs de niveaux différents en situation réelle. En leur proposant des exercices concrets, ils se seraient servis de la documentation en passant par l'index pour arriver à leurs fins, et il aurait suffi de relever les données mentionnées dans le tableau de la page précédente pour évaluer l'utilisabilité de l'index.

Cela étant, ces tests ont été bénéfiques à plusieurs égards. D'une part, interroger les ingénieurs m'a permis d'avoir différents points de vue sur les concepts définis dans le *Programmer's Manual*, de mettre à jour des différences terminologiques, et de profiter d'explications supplémentaires sur certaines notions que je ne dominais pas. D'autre part, les tests m'ont permis d'améliorer l'index à plusieurs niveaux.

4.3 incidences sur l'index

Malgré les difficultés rencontrées à l'élaboration des questionnaires et les doutes sur la valeur de leurs résultats, ils m'ont permis d'améliorer l'index :

- ▶ Modification de la formulation de certaines entrées qui ne donnaient pas d'informations assez claires sur le contenu des paragraphes marqués.
- ▶ Ajout de références croisées pour "corriger" le vocabulaire : certains ingénieurs utilisaient des termes différents pour certains concepts décrits dans le manuel ; il fallait donc les réorienter vers les termes adéquats.
- ▶ Ajout d'entrées/de sous-entrées : parfois je n'avais pas indexé certaines informations sous les termes auxquels les ingénieurs pensaient.

Conclusion

À l'heure à laquelle j'écris ce rapport, l'index du *Programmer's Manual* n'a pas encore été publié, et je suis encore en phase de test et de relecture. Cependant je crois pouvoir affirmer qu'il est à présent très proche de sa version définitive (cf. [annexe 8](#)). Il sera d'ici peu publié en interne, et pourra alors être soumis à l'appréciation du service assurance qualité et utilisé par ceux qui consulteront le manuel. Peut-être qu'alors je pourrais avoir des retours très concrets sur le travail que j'ai réalisé, puisque c'est sur ce point que l'indexation est la plus frustrante : sa qualité n'est pas facile à évaluer à première vue, et c'est en situation concrète qu'il doit être testé.

Chapitre 4

Relectures

En plus de la réalisation de l'index, j'ai été chargé de la relecture de certains documents en anglais, et j'étais invité à faire part de mes critiques à la lecture de la documentation de NexWave.

1 – Propositions d'améliorations

Mes connaissances limitées de la programmation et de la technologie de NexWave ne me permettaient pas de repérer d'éventuelles erreurs dans le contenu du *Programmer's Manual*. De la même manière, ma faible expérience dans la rédaction technique a été insuffisante pour que je puisse proposer des améliorations dans la forme du manuel. Du reste, son plan m'a semblé tout à fait cohérent : il suit la chaîne de production d'un système à base de composants, de leur codage à leur débogage. Certaines sections, cela dit, m'ont posé quelques problèmes à l'indexation parce que j'avais du mal à en déterminer le liant et que je les trouvais un peu "fourre-tout". Mais n'ayant pas mieux à proposer, la critique était là peu constructive.

Mes lectures répétées du *Programmer's Manual* m'ont permis cependant de mettre à jour certaines incohérences typographiques ou petites fautes. J'ai par exemple repéré l'utilisation hasardeuse de *NSI Runtime* et *NSI runtime*. Il s'est en fait avéré que le choix de mettre une majuscule ou non, au delà du service documentation, devait être fait en amont par les responsables de la partie commerciale de NexWave.

2 – Livret d'accueil

Par ailleurs, on m'a donné à réviser la version anglaise d'un livret d'accueil destiné aux nouveaux arrivants à NexWave et composé par la stagiaire en ressources humaines, vietnamophone d'origine.

Il s'agissait d'un assemblage de documents en anglais, textes de présentation de l'entreprise, traductions du règlement intérieur de l'entreprise, de la charte informatique, et autres documents administratifs circulant déjà depuis quelque temps.

Je ne me suis pas senti à l'aise pour réviser ce document : je me sentais en porte à faux par rapport à la ligne de conduite du Mastère qui se vérifiait totalement. En effet, les textes étant truffés d'erreurs, je me mettais à douter de mon propre anglais. D'autre part, il n'était pas évident de se situer par rapport au travail de relecture : la version française du document n'étant déjà pas de grande qualité, je me demandais si je devais y coller malgré tout, ou si je devais sacrifier le style de l'auteur au profit d'une ré-écriture de meilleure qualité (cf. [annexe 9](#)).

Ce travail de relecture a cependant été bénéfique dans le sens où il a été l'occasion de me replonger dans les règles de grammaire anglaise du *Chicago manual of style*.

3 – Documentation Eclipse

Dans le cadre de l'intégration du système DITA à la documentation, j'ai été chargé de réviser un ensemble de pages HTML constituant le système d'aide du logiciel de développement Eclipse adapté à la technologie NSI. Les textes étant de qualité, il s'agissait là surtout de repérer les coquilles ou les problèmes de mise en forme et de manque de cohésion typographique ou stylistique. Je n'ai en effet apporté que de très maigres modifications sur le contenu,

et quelques corrections orthographiques. J'ai pu appliquer dans cette mission les cours d'informatique appliquée suivis pendant le Mastère : j'ai effectué la plupart des corrections dans le code source des pages, en ajoutant ou en modifiant des balises HTML.

4 – « *Good written and spoken knowledge* »

Cette section est un peu hors-sujet dans le chapitre, mais je pense néanmoins qu'elle est liée aux tâches de relecture dont j'ai parlé précédemment. L'usage de l'anglais a été en effet prépondérant dans ce stage, et il convient d'expliquer pourquoi « l'entorse à la déontologie du Mastère », invoquée par ses responsables en acceptant que j'intègre NexWave, n'en est en fait pas une.

D'abord, l'anglais en rédaction technique ne doit être qu'un outil, un moyen pour faire passer une information, et surtout pas une fin en soi ou une manière pour l'auteur de montrer ses talents rédactionnels. C'est particulièrement vrai dans le cadre de NexWave où la documentation est essentiellement destinée à des non-anglophones natifs. La langue doit donc être d'une extrême simplicité, sans fioritures et idiotismes qui risqueraient d'assombrir l'information dont un programmeur a besoin pour réaliser une tâche. Évidemment, le texte doit être exempt de fautes d'orthographe et de grammaire, mais celles-ci peuvent aisément être évitées en consultant des sources sûres en la matière.

Ensuite, je suis convaincu que mon niveau en anglais était suffisant pour réaliser ma mission principale. La difficulté de compréhension des manuels de NexWave ne résidait en effet pas dans l'usage de la langue, mais bien dans le contenu technique. Et il est évident que la réalisation de l'index à proprement parler ne fait appel qu'à peu de talent d'écriture.

Par ailleurs, il me semble nécessaire de justifier ma participation à des missions de relecture. Comme je l'ai dit plus haut, les corrections que j'ai apporté aux documents techniques relus sont très minimales. Deux à trois personnes, francophones il est vrai, étant repassé sur les textes, on peut considérer que très peu de fautes de langue subsistent. On peut aussi considérer que les éventuels résidus passeront inaperçus aux yeux des programmeurs qui n'ont pas de formation spécialisée en langue, et dont la lecture n'est pas orientée-langue, mais orientée-information.

En revanche, dans le cas de la relecture de la version anglaise du livret d'accueil, il aurait clairement été préférable qu'elle ait été effectuée par un anglophone, puisque les qualités rédactionnelles étaient ici plus importantes. Cependant j'invoquerais dans ce cas une raison économique à ma réalisation de cette tâche : au vu des faibles enjeux que représentent ce livret d'accueil, il n'était pas nécessaire de faire appel à un des anglophones de NexWave qui ont des priorités autrement plus importantes, et un sous-traitant aurait nécessité une dépense de ressources sans doute conséquente et très discutable.

Enfin, l'argument éthique mis en relief dans la déontologie du Mastère et qui accuse les traducteurs francophones travaillant sur l'anglais de « voler » le poste d'anglophones natifs n'a pas raison d'être dans le contexte technique dans lequel j'ai réalisé mon stage. On m'a en effet dit à plusieurs reprises que le recrutement de personnes qualifiées à la fois en technique et en rédaction était difficile en raison du manque de main-d'œuvre.

Il me semble que ces raisons sont suffisantes pour justifier mon travail sur l'anglais pendant ce stage, malgré la déontologie du Mastère que je ne perds pas de vue, loin s'en faut.

Bilans

Une riche expérience, à bien des égards

De sa recherche à sa fin, ce stage m'aura appris dans bien des domaines. Il a constitué une conclusion tout à fait adaptée à mon parcours et à mon profil pluridisciplinaire, et a comblé ma curiosité et mon envie de découverte.

D'abord, ma recherche de stage m'a appris à garder mon sang froid et à persévérer dans la situation difficile dans laquelle je me trouvais à l'approche de l'échéance de juin pour commencer mon stage. D'une certaine manière, j'ai vécu en Irlande une situation comparable à celle du chômage, et le scénario tant redouté m'apparaîtra sans doute moins difficile si jamais il doit se présenter un jour. J'ai sans doute commis une erreur en choisissant de me rendre à Dublin sans être sûr d'y avoir un poste. Je ne m'y prendrai plus, et me rendrai dans les villes où des opportunités d'emploi m'attendront, et pas l'inverse. Ma recherche de stage m'a cependant permis de prendre connaissance de l'offre en matière de recrutement dans les domaines desquels le Mastère Rédacteur/Traducteur ouvre les portes. Il s'avère qu'elle est bien plus importante dans la rédaction technique que dans la traduction. De plus, j'ai pu avoir un aperçu des modes de recrutement et des qualités demandées, ainsi que des exigences des tests, et j'ai aussi pu me préparer aux entretiens d'embauche et aux contacts avec les recruteurs. Ma période dublinoise n'a pas été exempte de difficultés, mais a néanmoins été une expérience enrichissante, en matière de recherche d'emploi, et à niveau personnel surtout.

Ensuite, je suis ravi d'avoir pu travailler dans la startup innovante qu'est NexWave. Avoir un aperçu des méthodes de gestion à "l'anglo-saxonne", et d'une manière de travailler rigoureuse et normalisée m'a été bénéfique en sortie d'un cadre universitaire où l'organisation semble parfois un peu plus confuse. Le caractère international de l'entreprise m'a lui aussi plu. Si j'ai as-

sez peu parlé l'anglais durant mon stage, je l'ai toutefois souvent pratiqué à l'écrit, et entendu à maintes reprises dans la bouche des dirigeants ou des représentants de NexWave de passage à Montpellier. Je ne sais trop pourquoi, mais savoir que mon travail sera utilisé par des ingénieurs à l'autre bout de la planète me motive. Enfin j'ai l'impression d'avoir eu mon rôle à jouer dans le succès de la technologie proposée par NexWave. Mon index contribuera peut-être à faciliter l'usage de la documentation et des produits, et participera à l'établissement d'une bonne image de l'entreprise auprès de ses clients.

Travailler dans une entreprise essentiellement constituée d'ingénieurs en informatique m'a également permis de me rapprocher d'un domaine qui m'intéressait beaucoup il y a quelques années. J'ai pu au cours de ces mois acquérir quelques bases en programmation ainsi qu'une certaine culture informatique qui, j'en suis persuadé, me seront utiles dans ma carrière professionnelle. Que ce soit dans la traduction ou dans la rédaction technique, les candidats pouvant suivre les grandes lignes du code-source d'une application écrite en langage C ne sont sans doute pas légion, et mes quelques connaissances en la matière, qu'il conviendra d'approfondir, pourront représenter un avantage significatif pour ma candidature à un poste dans le futur.

Malgré ma mission particulière au sein du service documentation de NexWave, mes lectures approfondies des manuels existants m'ont permis d'assimiler un certain style d'écriture propre à la rédaction technique. Quand viendra le moment pour moi de rédiger des manuels d'instruction ou de référence, il me sera plus aisé de trouver les tournures appropriées. J'ai par ailleurs appris le fonctionnement d'un logiciel très utilisé en documentation technique, à savoir Adobe FrameMaker. À plusieurs reprises, j'ai retrouvé son nom dans des offres d'emploi, et je pense que la présence du nom du lo-

giciel dans mon CV pourra faire pencher la balance à mon avantage face à un autre candidat.

Enfin, ma mission très singulière m'a permis de mettre le pied dans un domaine tout à fait d'actualité dans le monde de la communication technique. L'indexation, ses enjeux et ses techniques font en effet l'objet de nombre de conférences, d'articles, d'études et de discussions dans la communauté des rédacteurs techniques. Non content d'avoir découvert qu'indexeur était un métier à part entière, j'ai pu m'initier aux pratiques de l'indexation, et me confronter aux difficultés des différentes étapes de la réalisation d'un index utilisable et de qualité. Je pense là encore qu'au vu de l'effervescence que provoque le concept d'index, mon expérience en la matière sera une force peu commune dans un CV.

Malgré des acquis certains, mon stage m'a aussi révélé un besoin de retour sur mon travail. Sans doute ce manque de confiance en mon travail s'estompera avec l'expérience, mais le travail d'indexation représente malheureusement un domaine où la qualité n'est pas évidente à évaluer. Lorsqu'un programmeur teste son programme, il peut obtenir des données précises sur son temps d'exécution, la mémoire utilisée, et ainsi évaluer sa qualité. L'index, en revanche, nécessite des tests d'utilisabilité plus difficiles à mettre en œuvre.

D'une manière générale, j'ai pu durant mon stage chez NexWave retrouver de nombreuses notions étudiées au cours de ma scolarité. De mon bac scientifique, d'abord, et de ses cours d'informatique et d'électronique. De LEA, aussi, et de son anglais appliqué. Et surtout du Master 2, de ses cours de rédaction technique, de certaines notions abordées en traductologie (j'ai adopté une double-approche, ascendante et descendante, pour réaliser mon index), d'informatique (le langage HTML), et de relecture de documents.

Comme nous avons pu le voir, mon stage chez NexWave a constitué une excellente conclusion à ma scolarité. Reste à savoir où mes diplôme et mon ex-

périence vont me conduire à présent. Malgré le fait que j'aie mis un pied dans un domaine plutôt porteur, j'aimerais vraiment avoir une expérience dans la traduction à niveau professionnel, mais aussi trouver à travailler dans langue de Cervantés !

Bibliographie

Trésor de la langue française informatisé. <http://atilf.atilf.fr/tlf.htm>

Wikipedia et Wiktionnaire (en langues française et anglaise)

Dictionnaires bilingues

Grand dictionnaire terminologique. www.granddictionnaire.com

Eurodicautom. <http://ec.europa.eu/eurodicautom/Controller>

Institut des sciences cognitives. http://dico.isc.cnrs.fr/fr/index_tr.html

Wordreference. www.wordreference.com/fr/

Programmation

ZHANG, Tony. *Le Langage C*, Paris, CampusPress France, 1999.

Le Site du zéro. www.siteduzero.com

Dictionnaire de l'informatique et d'internet. www.dicofr.com

Indexation et utilisabilité

The Chicago manual of style – 15th ed., University of Chicago Press, 2003.

MULVANY, Nancy C. *Indexing books*, University of Chicago Press, 1994.

BONURA, Larry S. *The Art of indexing*, John Wiley and Sons, Inc, 1994.

American Society of Indexers. www.asindexing.org

Society for Technical Communication – Indexing Special Interest Group. www.stcsig.org/idx

LATHROP, Lori. www.indexingskills.com

The Indexer, the international journal of indexing. www.theindexer.org

groupe de discussion TECHWR-L. www.techwr-l.com

Index Students community. www.indexstudents.com

groupe de discussion INDEX-L.

Utilisabilité.info. www.utilisabilite.info

HUGOT, François. *Une étude de l'utilisabilité de Cabri 3D*, Université Grenoble 1, 2005.

Annexes

Annexe 1. CV [40](#)

Annexe 2. Offre de stage [42](#)

Annexe 3. Tests de recrutement [43](#)

Annexe 4. Organigramme [44](#)

Annexe 5. Fenêtre principale du logiciel Adobe FrameMaker 6.0 [45](#)

Annexe 6. Programmer's Manual (extrait) [46](#)

Annexe 7. Questionnaires d'utilisabilité [47](#)

Annexe 8. Index réalisé [48](#)

Annexe 9. Livret d'accueil (extrait). [50](#)

Annexe 1. CV

Yoann Le Bleis

DOB : 04/08/1983

French student in Translation with strong computer skills and knowledge of a wide range of subjects, seeking work experience within a company that would valorise his profile.

DUBLIN 2,
Ireland

085 713 91 80

yoannlebleis@gmail.com

Education

- 2005-06 **Postgraduate Diploma in Technical Writing / Translation**
 ✓ *Communications, Localisation, Translation Studies, Linguistics, Terminology, Computer Science*
 ✓ *Participation in a fictitious translation agency managed by a team of 8 students under the supervision of a professional translator and an academic tutor*
- 2004 **Master's Degree in Multilingual Communication and Negotiation & Project Management (1st Year)** at University of Western Brittany in Brest
 1st semester : *tuition in Project Management, E-Marketing, Translation, Communications*
 2nd semester : *4 man team management of a project : study of the mechanisms and actors of cultural events in Brest, and organization of a jazz evening, then written report and public presentation*
- 2001-04 **3 years degree in LEA (Applied Foreign Languages)** at University of Western Brittany in Brest, and Jaime Ist University, Castellón, Spain
 subjects studied : *English and Spanish Languages and Cultures, Translation, Communications, Economics, Business Studies, Law, and Computer Science*
- 2001 **Baccalauréat S** (equivalent to Science A-Levels)
 subjects studied : *Physics, Chemistry, Mechanics, Electronics, Mathematics, English, Philosophy, History, Geography, and French Literature*

Work Experience

- Since Oct.04 Pizza delivery to finance my studies
- Sum. 2004 5-week training period with an independent sworn translator in Valencia, Spain (contact : Mr Vicente Albert Pamplò, vtalbert@vtalbert.com)
 ✓ *translated various types of documents into French, English and Spanish*
 ✓ *revised French translations*
- Q1 2003 **Devised a questionnaire** for the *French Building Federation* in order to study 1st degree pupils' interest in jobs in the sector
- 1999-02 Accompanied and helped run activities for groups of children and teenagers as a camp counsellor

Proficiencies

Language Skills

| | |
|---------|--|
| French | Mother tongue |
| English | Good written and spoken knowledge |
| Spanish | Good written and spoken knowledge <i>7 months spent in the Valencia region of Spain through the Erasmus programme</i> |
| Italian | Basic knowledge |

Computer Skills

Self computer educated, practising for amusement and interested in the subject for about **10 years**.

Comfortable with *MS Windows, MacOSX (iBook owner), Internet, MS Office Applications, CAT programmes (Trados, Similis)*

Strong bases on **HTML language** and website creation programmes (*MS Frontpage*)

Basic knowledge of image manipulation programmes (*Photoshop, Gimp*) and DTP programmes (*Adobe InDesign*)

Miscellaneous

BAFA **Camp councillor diploma** (a 4-week course including a 2-week training period in a teenagers camp)

Driving Licence

Annexe 2. Offre de stage



Stagiaire en rédaction technique (H/F)

NexWave Solutions (31 collaborateurs), société engagée dans le développement et la production d'une technologie à base de composants pour le marché de l'embarqué, recherche un(e) stagiaire pour son service documentation pour une durée de 3/4 mois :

Rattaché(e) à la rédactrice technique, vous assurerez:

- L'indexation de la documentation technique existante : identification des concepts clés, hiérarchisation, marquage et génération des tables d'index.

Outils : FrameMaker, Microsoft Office, outils internes et plugins.

Domaine technique : informatique, systèmes embarqués, programmation.

Vous possédez impérativement un bon niveau d'anglais.

Rigueur, esprit de synthèse et goût pour le travail d'équipe vous permettront de réussir.

Lieu de travail : Montpellier.

Date de démarrage : à convenir

Contact : Céline CAMPI, hr@nexwave-solutions.com

NexWave Solutions

1068, rue de la Vieille Poste

CS 89006

34967 Montpellier Cedex 2

Tel : 04 99 52 89 89 / Fax : 04 99 52 80 38

Visit us at : <http://www.nexwave-solutions.com>

Annexe 3. Tests de recrutement



1 Translation

Translate the following paragraph in English.

The research project on a software based on sharing knowledge components aims at developing software engineering techniques applicable to creation and evolution of Web applications. Although Web techniques profit from several years of research and practice in software engineering, new priorities and technological changes contribute to the unique feature of the techniques used in Web applications engineering. (...)

Le projet de recherche sur un logiciel à base de composants servant au partage des connaissances vise à mettre au point des techniques de génie logiciel pouvant s'appliquer à la création et à l'évolution d'applications Web. Bien que les techniques Web tirent profit de plusieurs années de recherche et de pratique dans le secteur du génie logiciel, les nouvelles priorités et les changements technologiques contribuent au caractère unique des techniques du génie utilisées pour les applications Web. (...)

2 English proofreading

Correct the English and style errors in the following extract from a NexWave user manual.

5 Exported and imported interfaces

An interface is described by its specification which contains attribute descriptions, function prototypes, and additional information such as parameter descriptions, usage constraints, and limitations. The specification enables the exporter component to implement (and then providing) the interface and the importer component to request (and then using) the interface.

Within the frame of the specification, interface definitions are flexible:

- Components can export only a subset of the possible functions and attributes, therefore providing a limited implementation of the interface.
- Components can import only a subset of the possible functions and attributes, if they need only a part of the interface's functionalities. (...)

3 Technical writing

Edit and re-write (if necessary) the following steps to obtain two clear and crisp procedures: "To highlight, cross out or underline text" and "To delete a highlight, cross out, or underline markup." You may use the back of this page.

Highlighting, crossing out, and underlining text

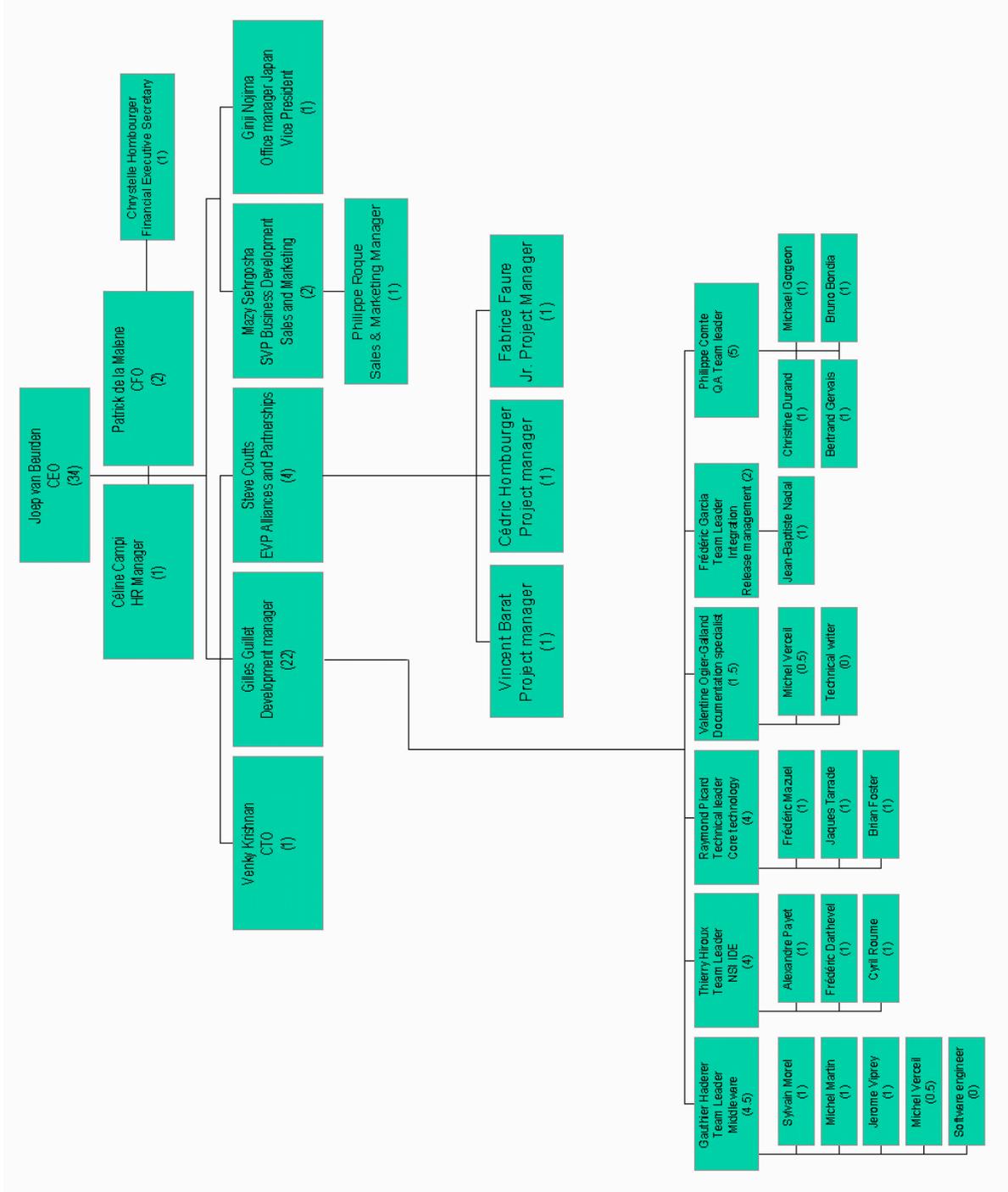
You can use the Highlight Text tool, the Cross-Out Text tool, and the Underline Text tool in order to add comments to an Adobe PDF document.

To highlight, cross out, or underline text

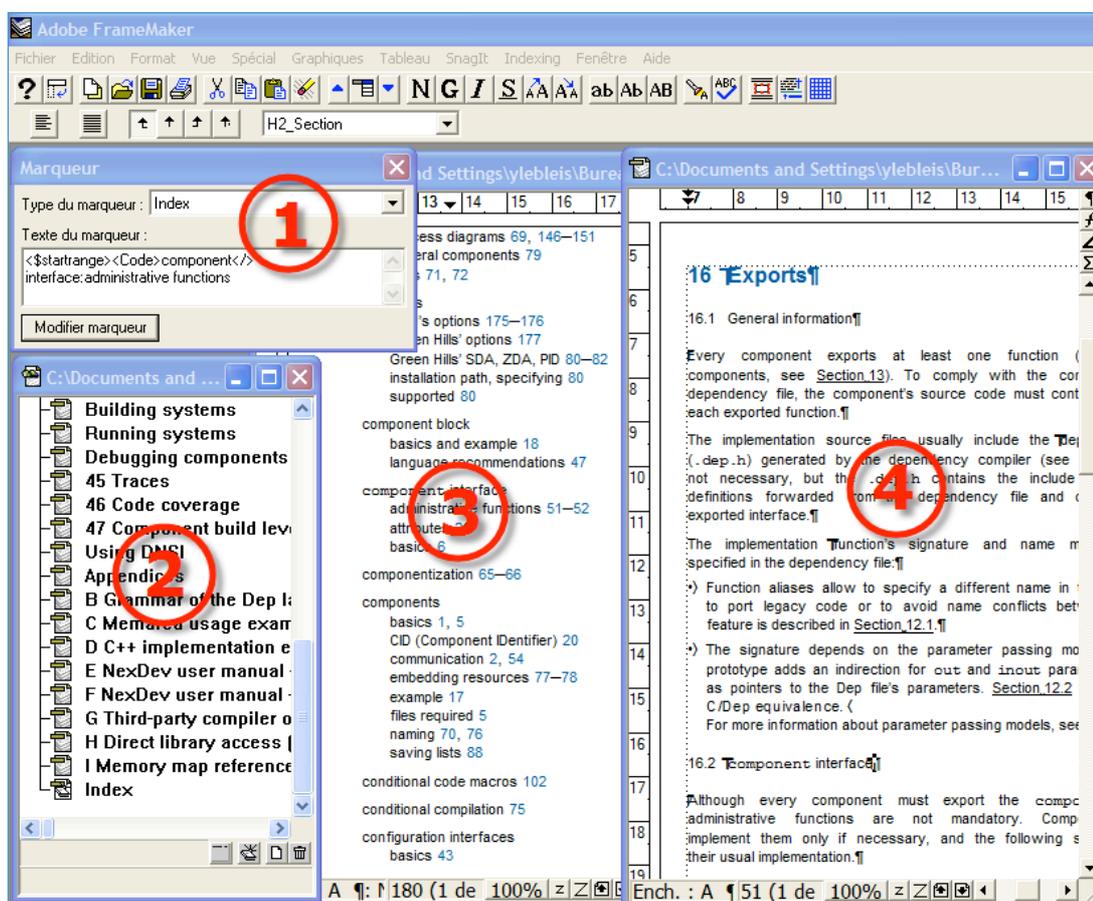
Note: you can mark up a rectangular area of text by pressing the `Ctrl` button while dragging text.

- 1 On the "Comments & Markup" toolbar,
- 2 Click "Commenting Tools" > "Highlighting",
- 3 Choose the desired tool,
- 4 Drag from the beginning of the text you want to mark up. (...)

Annexe 4. Organigramme



Annexe 5. Fenêtre principale du logiciel Adobe FrameMaker 6.0



- 1. Fenêtre des marqueurs :** on y indique le type du marqueur et le texte qui sera indexé, avec des balises de mise en forme ou diverses options pour le classement alphabétique, la présence ou l'absence du numéro de page, etc.
- 2. Fenêtre livre :** comporte la liste de tous les fichiers qui composent le livre, et qui correspondent souvent aux chapitres. L'index est volontairement placé en fin de livre.
- 3. Index** généré automatiquement à partir des marqueurs de type index, des pages références qui définissent les règles de mise en forme, et des balises insérées dans le texte des marqueurs.
- 4. Texte-source** dans lequel on place les marqueurs (symboles T).

Annexe 6. Programmer's Manual (extrait)

Describing components & interfaces

NSI Programmer's Manual

Component

8 Component

Each component exports the `component` interface, which identifies the component within the NSI system.

```
exported interface component {

    /* identification of the interface */
    const nsi_uuid_t iid = "204f0640-5c25-4f80-8f3e-08c0ef562ba2";

    /* identification of this component */
    const nsi_uuid_t cid = "00000000-0000-0000-0000-651d37ef5efe";

    /* human-readable name and description */
    const nsi_string_t name = "antivirus03";
    const nsi_string_t descr = "self-updating antivirus example";

    /* build version */
    const nsi_version_t build = MAKE_VERSION (1, 0, 0);
};
```

The above example is the minimal export of the `component` interface, which also contains four administrative functions that are only exported when necessary. The interface is documented in the [NSI Reference Manual](#); here is a quick overview of its attributes:

- The `iid` attribute is the `component` interface's identifier: it is identical for all the components.
- The `cid` attribute uniquely identifies the component: you generate it at component creation, either with the `uuidgen` command-line utility or the Eclipse wizard.
- The `name` attribute appears in NexBuilder and in debug traces. It is also the name of the component's binary file. An empty string is not allowed.
- The `descr` attribute is a short, meaningful overview appearing in NexBuilder and helping the system builder to select the component. An empty string is allowed.
- The `build` attribute is constructed from the `MAKE_VERSION` macro with major, minor and micro version numbers (see [Section 10.4](#)). This attribute tracks the evolutions of the component's implementation.

A component can export any combination of the four administrative functions, depending on its implementation. A short overview of these four functions is provided in [Section 16.2](#).

The `component` interface requires to include `<nsi/component.h>` in the dependency file.



Annexe 7. Questionnaires d'utilisabilité

| Questions | concepts pensés* |
|--|---|
| vous voulez compiler plusieurs composant à la fois | <u>compilation</u> ; NexDev ; nex-make |
| vous voulez insérer la police/image dont un composant a besoin directement dans son binaire | <u>fonts</u> ; <u>images</u> ; GUI ; NexGUI ; internationalization ; embedding resources ; Makefiles |
| vous écrivez un fichier Dep et ne vous souvenez plus des mots-clés définissant un import | <u>dependency files</u> |
| vous voulez avoir des informations sur les pointeurs que contient la vue | views ; pointers |
| vous êtes en phase de test d'un composant et voulez qu'il vous "parle" | traces ; NexTerminal ; console ; terminal ; compilation ; debug ; component |
| vous voulez écrire un composant en C++ et savoir si NSI restreint les possibilités de ce langage | compilation ; <u>C++</u> |
| vous voulez changer le nom du .dep (composant "toto", dependency.dep et pas toto.dep) | <u>dependency files</u> ; naming ; Makefiles |
| vous voulez utiliser un raccourcis pour importer directement une fonction | shortcuts, aliases |
| vous voulez adapter un logiciel monolithique à la technologie NSI | componentization ; converting ; adapting ; <u>porting</u> ; <u>refactoring</u> ; legacy code |
| vous voulez contrôler l'utilisation par une application des zones de mémoire de la plate-forme cible | memory ; NexBuilder |
| vous voulez utiliser votre composant avec votre librairie | libraries ; Direct Library Access |

* en gras, les entrées principales présentes avant le test ; soulignées, les entrées ajoutées par la suite

Annexe 8. Index réalisé (extrait)

NSI Programmer's Manual

Index

A

- administrative functions 14–15, 51–52
- aggregate types 32
- aliases
 - functions 41
 - interfaces 63–64
- arrays
 - C/Dep equivalence 34
 - usage 33
- assertions 102
- attributes
 - as import constraints 37
 - component interface 20
 - interface (possible types) 24

B

- bit-mask operator 39–40
- boolean type 26
- build levels 112
- build options for several components 79

C

- C++. See EC++. languages
- cardinality (import types) 12
- CBSE (Component-Based Software Engineering) 1, 86
- CID (Component Identifier)
 - basics 20
 - uuidgen 164
- code coverage
 - basics 105–106
 - creating systems for 107–108
 - nExecute (usage example) 111
 - results, manipulating 109–110
- code generated
 - indices 56
 - views 55

- command line
 - compilation 71
 - image generation 91
 - NSI image generation 158
- compatibility
 - components 3
 - configurations supported 122
- compilation
 - basics 69
 - build levels 112
 - conditional 75
 - custom sections 83
 - languages supported 53
 - options (See also debug. optimization) 174
 - process diagrams 69, 146–151
 - several components 79
 - tools 71, 72
- compilers
 - GNU's options 175–176
 - Green Hills' options 177
 - Green Hills' SDA, ZDA, PID 80–82
 - installation path, specifying 80
 - supported 80
- component block
 - basics and example 18
 - language recommendations 47
- component interface
 - administrative functions 51–52
 - attributes 20
 - basics 6
- componentization 65–66
- components
 - basics 1, 5
 - CID (Component Identifier) 20
 - communication 2, 54
 - embedding resources 77–78
 - example 17
 - files required 5
 - naming 70, 76
 - saving lists 88
- conditional code macros 102
- conditional compilation 75

- configuration interfaces
 - basics 43
 - example 44
 - naming conventions 43
 - configurations supported by NSI 122
 - conflicts, naming. *See* aliases
 - console. *See* command line
 - constraints, import
 - See also* import types
 - basics 36
 - bit-mask operator 39–40
 - functions and attributes 37
 - IID (Interface Identifier) 36
 - operators allowed 36
 - relational 38–40
 - scalar attributes 39
 - string attributes 38
 - container header files 46–47
 - CPU models (NexDev names) 122
 - cycles, import 16
 - CygWin 71
- ## D
- debug
 - activating at compilation 98
 - assertions 102
 - build levels 112
 - code coverage. *See* code coverage
 - compilation diagram 150
 - conditional code 102
 - formats 80
 - image 99
 - memory poisoning 114
 - parameter validity checks 112–114
 - traces. *See* traces
 - debuggers (NexDebug, gdb, ICE) 100
 - .dep files. *See* dependency files
 - .dep.h files. *See* dependency header files
 - Dep language grammar 123–132
 - dependencies
 - basics 2
 - resolution 13
 - dependency compiler
 - code generated 53–56
 - files generated 45
 - languages supported 53
 - dependency files
 - basics 18–19
 - changing default name 76
 - example 17
 - import types keywords 23
 - language recommendations 45–48
 - optimizing at compilation 83–84
 - dependency header files
 - basics and contents 48
 - example (generated code) 55
 - including in source files 51
 - Dependency Manager
 - administrative functions 14–15
 - basics 2, 13
 - depmsgz interface 13
 - import structures 53
 - runtime imports 61
 - views 54
 - Devel build level 112
 - distribution (import types) 12
 - Distribution Manager
 - basics 115
 - T-Integrator 121
 - DLA (Direct Library Access) 96, 178
 - DNSI (Distributed NSI)
 - basics 3, 115–116
 - conditions of use 117–118
 - import type (local/remotable) 12
 - passing models, parameter 25
 - serial transport configuration 119–120
 - T-Integrator 117, 121
 - dynamic imports. *See* runtime imports
 - dynamism (import types) 10–11
- ## E
- EC++ (Embedded C++)
 - example 138–145
 - file extensions 73
 - usage 67–68
 - Eclipse 72

Annexe 9. Livret d'accueil (extrait)

Avant révision :

1. Year-end interview

Every year in November, the employee and his line manager will meet each other within a context of a year-end interview in order to facilitate the communication between evaluator and evaluated, to get the feedback relating to employee's work and to understand his strong and weak points.

This yearly interview is divided into two parties: appraisal interview and professional interview.

a/ Appraisal interview consists of reviewing the performance regarding the job occupied by employee and his personal performance, evaluating his collective and individual objectives in the current year and determining them for the coming year as well as discussing about his wishes in term of internal and geographical mobility.

b/ Professional interview: after the appraisal interview, the professional interview is the essential way for employee to join our training program. This interview is devoted to evaluate his trainings during current year and to collect his training needs for the coming year on the part of both employee and line manager. This interview also reminds the staff of the existence of DIF and the opportunity to use it.

Après révision :

1. Year-end interview

Every year in November, a one-to-one meeting is organized between each employee and his/her line manager in order to facilitate the communication between the evaluator and the evaluated, to get the feedback relating to the employee's work and to understand his/her strengths and weaknesses.

This year-end interview is divided into two parts: the appraisal interview and the professional interview.

a/ The goal of the appraisal interview is to evaluate the efficiency and the performance of the employee. Moreover the completion of both collective and individual objectives is examined for the current year, and new objectives are determined for the coming year. Finally, the employee's wishes in terms of internal and geographical mobility are discussed.

b/ Completing the appraisal interview, the professional interview is devoted to the employee's training program. During this interview are evaluated the training sessions the employee attended in the current year, and are collected his/her wishes for the coming year, as well as his/her line manager's. This interview also reminds the staff of the existence of the DIF and the opportunity to use it.