Le Format HL7

HL7 ?

- HL7 est un format de fichier, un projet, mais aussi une « NORME » d'échanges d'informations dans le domaine médicale, on parlera plutôt de spécification HL7.

- Ce n'est pas le seul format d'échange dans le monde médical, il existe d'autres normes comme Hprim, pivot, pn13, EDI, edifact, HprimXml, B2 Noémie, etc…

- HL7 devient l'un des formats les plus utilisé dans les hôpitaux français et internationaux(100%).

- Il existe 120 types de messages différents en HL7, qui ont chacun leur spécificités et leurs domaines d'utilisations.

Les types les plus utilisés (non exhaustif)

HL7 ADT => qui permet d'échanger les identités et les mouvements des patients

HL7 ORM => qui permet d'échanger les prescriptions de tout types (médicament, radio, laboratoire, etc…)

HL7 ORU => qui permet d'échanger les résultats des examens, soit sous formes de valeurs discrètes, ou de texte, ou d'un lien d'un fichier associé.

HL7 RDE => qui permet d'échanger les prescriptions spécifiques vers le domaine de la pharmacie (c'est un ORM spécialisé pharmacie.

HL7 SIU => qui permet d'échanger les informations de rendez vous (échanges entre agenda)

HL7 MDM => qui permet d'échanger les informations de documents textes, sous forme d'une url d'un document attaché ou intégré dans le message et codé en base64.

Notions de versions en HL7

Hl7 au fils des années s'est décliné dans différentes version et ont apporté chacune des améliorations fonctionnelles

Les versions existantes :
2.1, 2.2, 2.3, 2.3.1, 2.4, 2.5, 2.5.1, 2.6, 2.7, 2.8, 2.9, FHIR

Les versions les plus utilisé en France sont les version 2.3.1 et 2.5.1

Il existe également des versions arrangées à la « sauce française » pour prendre en compte certaines spécificités française de prise en charge des patients (numéro ins-c, information de SS, etc…)
Et des segments spécifiques à la France comme les segments ZRE, ZFU, Zxx…

On trouve également des dérivent que l'on nomme HL7 PAM IT 30 & IT 31 qui permettent de mettre en place une notion de bidirectionnalité notamment dans le domaine d'HL7 ADT

Structure d'un fichier HL7

```
MSH|^~\&|SA_AMCK|SF_MCK|DXSRA|Resultat labo|20161026082508|A05|ADT^A01|19105586|P|2.3.1||||||8859/1|\r
EVN|A01|20161026070000||||\r
PID|1||497406^^^IF_MCKN||BLEUSE^BENEDICTE^^^^^L~LEFEVRE^BENEDICTE^^^^^M|LEFEVRE|19640726000000|F|||97  RUE
KEYWORTH^^FEIGNIES^^59750^100|100|06.72.95.82.60 MARI|||M||686303855|2640759392148||||MAUBEUGE|||||100^100||N\r
PV1|1|I|1058^1217^217|R|||||||||||||686303855|||||||||||||||||||||||20161026070000||||||7250630|A|\r
PV2|||^^^8^^|||||||||||||||||||N\r
ZFU|1058|20161026070000|2601|20161026070000|1058|20161026070000\r
ZRE||||59600|||62577||\r
```

Un message HL7 est formé de segments => 1 segment = 1 ligne de caractères (fin de ligne est le caractère \r)

Un segment est un groupe de 3 caractères majuscules au début d'une ligne (PID, PV1, …)

Il existe en HL7 environ 150 type de segments différents.

Chaque type de messages HL7 possède ses propres segments.

Dans les messages HL7 on trouve des segments commun à tous les types de messages HL7
(MSH, EVN, PID, PV1, PV2, AL1, NK1)

D'autres segment sont spécifiques au type du message (ici il faut lire les spécifications pour les connaitre)

Structure d'un segment HL7

```
PID|1||497406^^^IF_MCKN||BLEUSE^BENEDICTE^^^^^L~LEFEVRE^BENEDICTE^^^^^M|LEFEVRE|1
9640726000000|F|||97  RUE KEYWORTH^^FEIGNIES^^59750^100|100|06.72.95.82.60
MARI|||M||686303855|2640759392148|||MAUBEUGE|||||100^100||N\r
```

Un segment commence toujours par un groupe de 3 caractères qui indique le nom du segment.
Un segment est formé de champs qui comporte des données séparées en général par le signe pipe « | ».

Un champ est l'ensemble des caractères qui se trouvent entre 2 caractères pipe « | ».

Un champ peut être composé de sous champs qui sont séparés par le signe « ^ »
97 RUE KEYWORTH^^FEIGNIES^^59750^100

Un sous champ peut être composés de sous-sous-champs, chaque s/s/champs sont séparés par le caractère « ~ »
L~LEFEVRE

Etc… Jusque 4 sous niveaux…

Chaque champ est numéroté à partir de la valeur 1 qui commence après le nom du segment, on nomme toujours un champ par son NOM-num
Exemple PID-3 = 497406^^^IF_MCKN

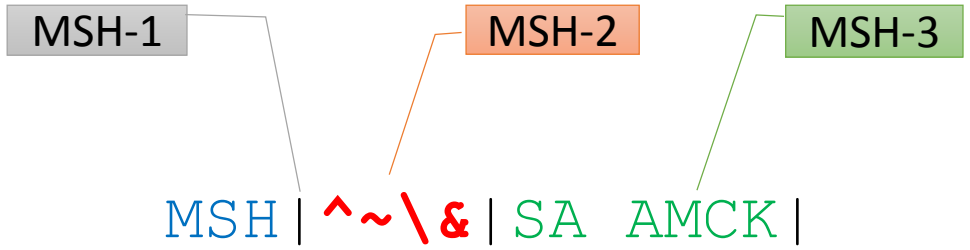
De même on numérote les sous-champs et les sous-sous-champs à partir de la base 1 et on sépare ces numéro par un « . »
Exemple : PID-5=BLEUSE^BENEDICTE^^^^^L~LEFEVRE^BENEDICTE^^^^^M
PID-5.7=L~LEFEVRE
PID-5.7.2=LEFEVRE

Un segment particulier « MSH »

```
MSH|^~\&|SA_AMCK|SF_MCK|DXSRA|Resultat labo|20161026082508|A05|ADT^A01|19105586|P|2.3.1||||||8859/1|\r
```

Le segment MSH est particulier dans le sens ou il définit les caractères de séparations qui seront utilisés dans le message

- Séparateur des champs
- Séparateur des sous champs
- Séparateur des sous/sous champs N3 & N4



En général, on utilise les séparateurs par défaut | ^ ~ \ &

De ce fait, on compte l'ordre des champs comme ceci :

MSH-1 est le séparateur des champs
MSH-2 les séparateurs de sous champs (4 sous niveaux possible), il est à noter que ces caractères ne pourront pas être utilisés dans la valeur des champs. Il faudra dans ce cas coder ces caractères en Hexadécimal 0xB3 0x5E 0x7E 0x5C 0x26

Notions de typage des champs et tables de valeurs

Chaque champs d'un segment HL7 est défini par un type qui cadre les bornes des valeurs admissibles dans ce champ.

Un type peut être une chaine de caractères (ST), un entier (ID), une date (DT), une structure complexe (XCN) qui est composé de types simples.

Certains champs n'acceptent qu'un ensemble de valeurs fini, c'est ce que l'on nomme les tables de valeurs HL7, il existe des tables pour chaque type de champs
Exemples table 0007 : Admission Type

VALUE	LABEL
A	Accident
E	Emergency
L	Labor and Delivery
R	Routine

SEQ	LENGTH	DT	OPT	TBL #	NAME
XCN.1	0	ST	O		Id Number
XCN.2	0	FN	O		Family Name & Last Name Prefix
XCN.3	0	ST	O	FirstName	Given Name
XCN.4	0	ST	O		Middle Initial Or Name
XCN.5	0	ST	O		Suffix
XCN.6	0	ST	O		Prefix
XCN.7	0	IS	O		Degree
XCN.8	0	IS	O		Source Table
XCN.9	0	HD	O		Assigning Authority
XCN.10	0	ID	O		Name Type Code
XCN.11	0	ST	O		Identifier Check Digit
XCN.12	0	ID	O		Code Identifying The Check Digit Scheme Employed
XCN.13	0	IS	O		Identifier Type Code
XCN.14	0	HD	O		Assigning Facility
XCN.15	0	ID	O		Name Representation Code

ACK HL7

Le récepteur d'un message doit envoyer un accusé de réception applicatif ou simplement de transmission pour valider l'échange.

Cela consiste en l'envoi d'un message HL7 de type ACK sur la même socket et selon la méthode décrite ci-dessus.

L'ACK accuse de la bonne réception du message ainsi que de son traitement dans le cas applicatif.

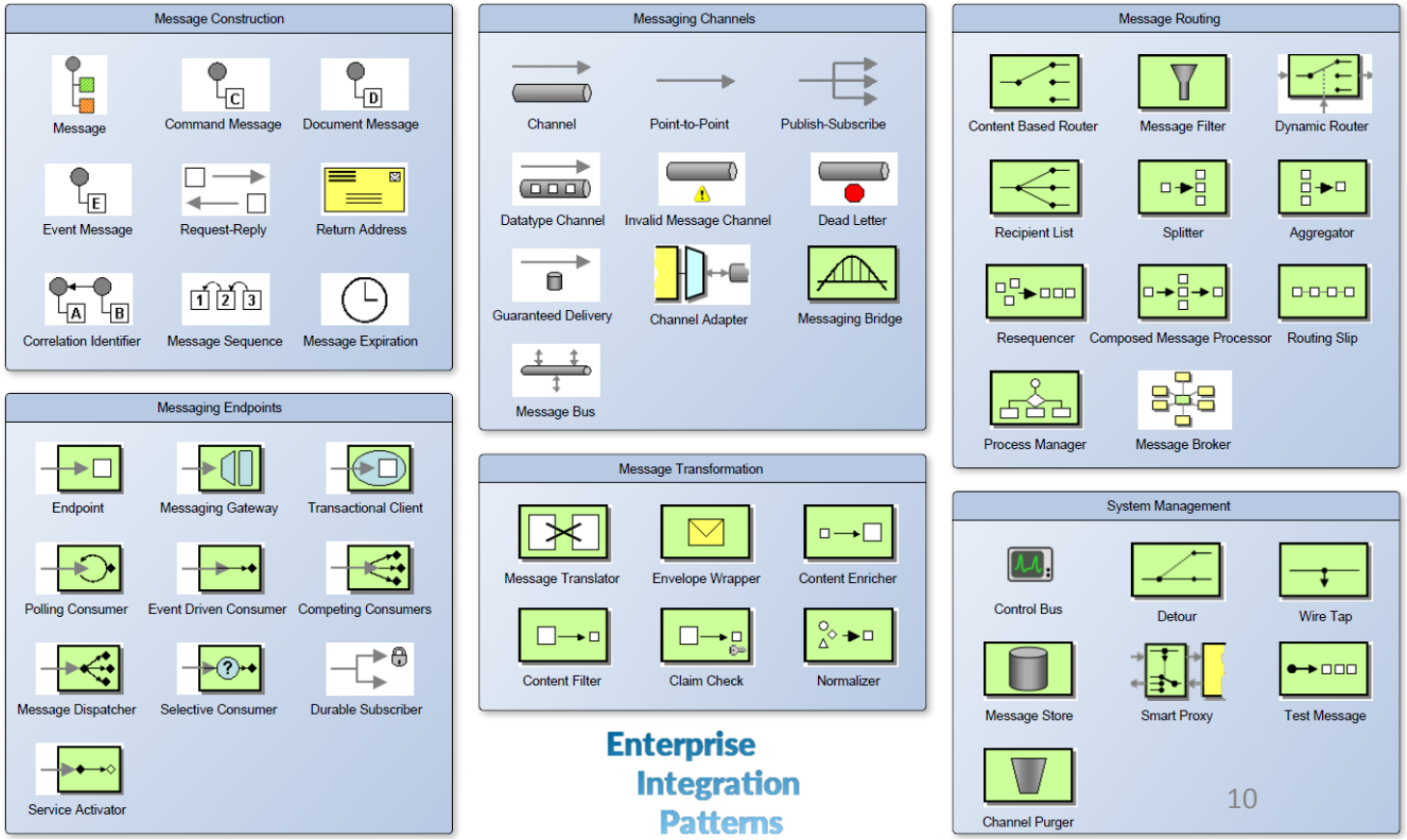
MSH|^~\&|Millenium|CHV|SA_AMCK|SF_MCK|20161026131726|S01|ACK^A16|19109295|P|2.3.1|||||8859/1|
MSA|AA|19109295

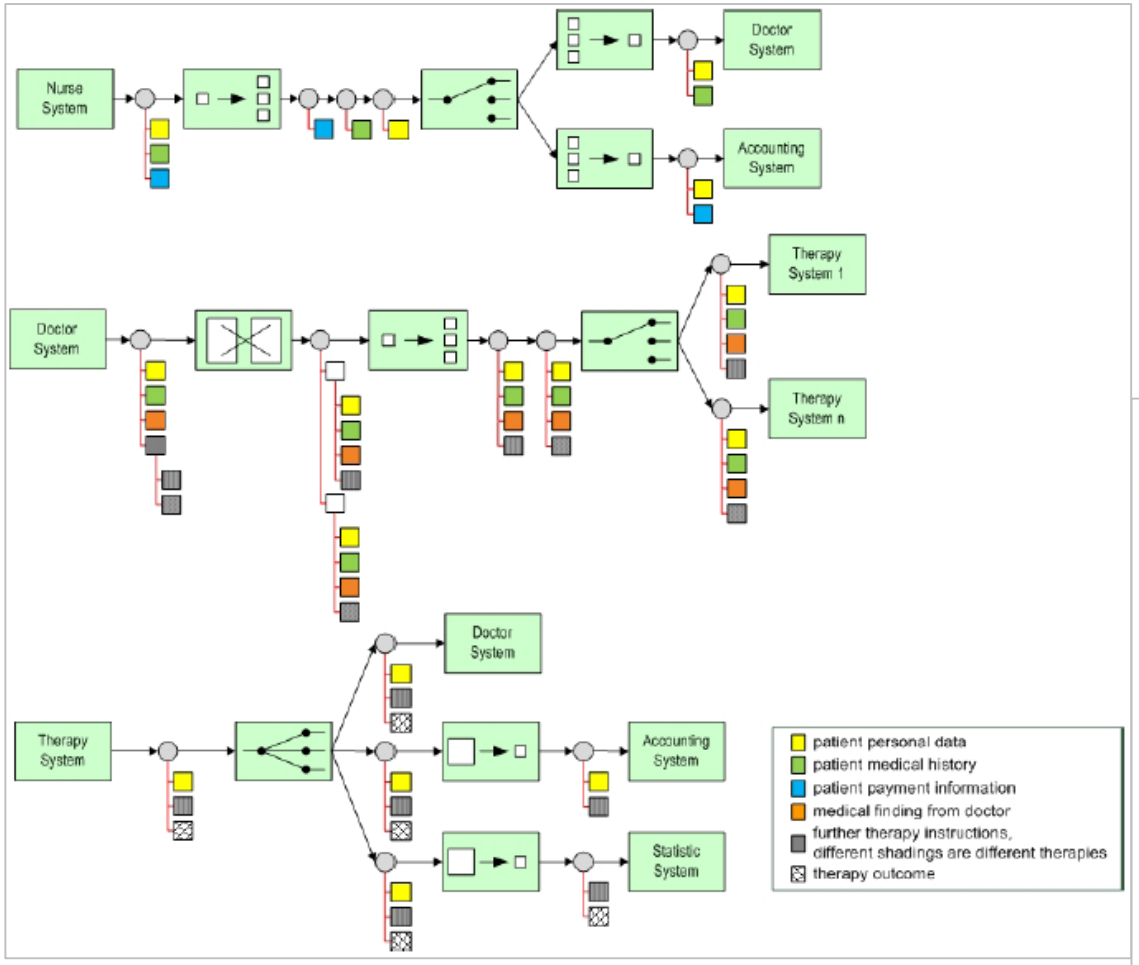
VALUE	LABEL
AA	Original mode: Application Accept - Enhanced mode: Application acknowledgment: Accept
AE	Original mode: Application Error - Enhanced mode: Application acknowledgment: Error
AR	Original mode: Application Reject - Enhanced mode: Application acknowledgment: Reject
CA	Enhanced mode: Accept acknowledgment: Commit Accept
CE	Enhanced mode: Accept acknowledgment: Commit Error
CR	Enhanced mode: Accept acknowledgment: Commit Reject

Notions sur l’ « Enterprise Integration Patterns »

Les bases de l’interopérabilité ont été établis par Gregor Hohpe & Bobby Woolf et formalisé en 2003 lors de la publication de la thèse et du livre qui à suivi cette thèse sur l’ Enterprise Integration Patterns (Design, Building, and deploying Messaging Solutions)
Cette thèse jette les bases de la notions de messaging Endpoints, de message Construction, de Channels de message Routing, de message Transformation et system management.

Messaging Endpoints = points d’entrés et de sorties des messages.
Messaging Channels = transport et traitement des messages.
Message routing = Routage des messages
Message transformation = transformation des messages.
Messaging construction = construction de nouveau messages ou augmentation de la signification d’un message.
System management = supervision des flux de messages.





Le langage symbolique mis en place par Gregor Hophe & Bobby Woolf, permet de modéliser l'ensemble des flux d'un système de gestion de messages inter-applicatif.