

I think that the variety of patron information kept in individual systems is such that it will be very difficult to come up with a canonical patron record to pass between systems. I have therefore loosely modelled this external on the Z39.50 GRS syntax, with the intention that there should be some smallish set of "standard" elements most systems will recognize, and the rest can be transferred as labelled elements. The patron record is therefore a recursive list (sequence) of type-content pairs. There is an enumeration of basic types here and other data can be transferred using a string label for the content. The data types for the content element have been taken from the base ILL standard.

The advantage of using a recursive list is that it allows a system to place data at the level of granularity that it uses, or that has been agreed to within a consortium.

It would be possible to make the tags an external tagSet, as with GRS, so that new tag sets can be registered with the Registration Authority, but this seems a bit like overkill.

Since this approach can be a bit hard to follow, I have given two examples (using the formal ASN.1 value notation, but omitting the tags) at the end which I hope will make the kinds of record you get somewhat easier to understand. The first example shows a record with data at a very coarse level of granularity. The second record includes the same data, but has much finer level of granularity, so that the system can easily determine, e.g. if a telephone call is local or long distance

ILL-Supplemental-Client-Info DEFINITIONS ::=

BEGIN

IMPORTS ILL-String, System-Id, System-Address, Postal-Address from
{iso standard 10161 abstract-syntax ill-apdus};

SupplementalClientId ::= SEQUENCE OF ClientInfoData

ClientInfoData ::= SEQUENCE {
 infoType ClientInfoType,
 infoContent [3] IMPLICIT SEQUENCE OF ClientInfoContent
}

ClientInfoType ::= CHOICE {
 standard [1] IMPLICIT Standard-Client-Info-Type,
 local [2] ILL-String,
}

ClientInfoContent ::= CHOICE {
 stringContent [1] ILL-String,
 postalAddress [2] IMPLICIT Postal-Address,
 nameOrId [3] IMPLICIT System-Id,
 statusCode [4] IMPLICIT StatusCode,
 definedNumber [5] IMPLICIT INTEGER,
 nested [6] IMPLICIT ClientInfoData
}

```

Standard-Client-Info-Type ::= ENUMERATED {
-- tags for hierarchical levels :
    idInfo           (0),
    nameInfo         (1),
    contactInfo      (2),
    statusInfo       (3),
    otherInfo        (4),
    homeAddressInfo  (5),
    workAddressInfo  (6),
    faxNumber        (7),
    voiceNumber      (8),
    pagerNumber      (9),
    eMailInfo        (10),
-- tags for id:
    systemIdNo       (21),
    socialSecurityNo (22),
    accountNo        (23),
    driversLicenceNo (24),
    other            (25), -- will require contents with a string tag
-- tags for names:
    wholeName        (30),
    lastName         (31),
    firstName        (32),
    initials         (33),
    prefix           (34),
    suffix           (35),
-- tags for addresses:
    name             (40),
    streetAndNumber  (41),
    additionalInfo   (42), -- e.g. Apt., Floor
    boxAddress       (43),
    city-town        (44),
    state-prov-region (45),
    country          (46),
    postalCode       (47),
-- tags for dialling info:
    telephoneCountryCode (50),
    telephoneAreaCode   (51),
    telephoneLocalNumber (52),
    telephoneExtension   (53),
-- tags for email:
    eMailService        (60),
    eMailAddress        (61)
}

```

```

StatusCode ::= ENUMERATED {
    undergraduate (0),
    graduate      (1),

```

```

faculty          (2),
staff           (3),
visitor         (4),
generalUser     (5),
juvenile       (6),
VIP            (7),
subscriber      (8),
suspended      (9),
other          (10),
unspecified    (11)
}

```

-- additions to this list may be made from time to time
-- and will be registered with the ILL Registration Authority

END

Examples (each new SEQUENCE or SEQUENCE OF type value begins on a new line):

A display format of the record is:

```

Patron Number: 123456789
Name:         Smith, John
HomeAddress:  123 Main Street
              Smallville ON
              K0A 1A0
              voice: (613) 923-4567
Status:       Undergraduate

```

1. A record with data at a minimal level of granularity

```

{ -- the initial unidentified SEQUENCE OF
  {infoType standard systemIdNo,
    infoContent -- a SEQUENCE OF
      {stringContent "123456789"}  },
  {infoType standard nameInfo,
    infoContent
      {stringContent "Smith, John"}  },
  {infoType standard homeAddress,
    infoContent
      {stringContent "123 Main Street, Smallville ON, K0A 1A0"}},
  {infoType standard voiceNumber,
    infoContent
      {stringContent "(613) 923-4567"}  },
  {infoType standard status,
    infoContent
      {statusCode undergraduate}  }  }

```

2. A record with the same data at a high level of granularity

```
{
  {infoType standard idInfo,
  infoContent
    {nested
      {infoType standard systemIdNo,
      infoContent
        {stringContent "123456789"} } } },
  {infoType standard nameInfo,
  infoContent
    {nested
      {infoType standard lastName,
      infoContent
        {stringContent "Smith"} },
      nested
      {infoType standard firstName,
      infoContent
        {stringContent "John"} } } },
  {infoType standard contactInfo,
  infoContent
    { nested
      {infoType standard homeAddress,
      infoContent
        {nested -- this could have been a postalAddress choice instead
          {infoType standard streetAndNumber,
          infoContent
            {stringContent "123 Main Street"} },
          nested
          {infoType standard city-town,
          infoContent
            {stringContent "Smallville"} },
          nested
          {infoType standard state-prov-region,
          infoContent
            {stringContent "ON"} },
          nested
          {infoType standard postalCode},
          infoContent
            {stringContent "K0A 1A0"} },
          nested
          {infoType standard voiceNumber,
          infoContent
            {nested
              {infoType standard telephone-area-code,
              infoContent
                {stringContent "613"} },
```

```

        nested
        {infoType telephone-local-number,
         infoContent
          {stringContent "923-4567" } } } } } } } },
{infoType standard status,
 infoContent
  {statusCode undergraduate} } }

```

Alternative Specification for the Record. This one has everything explicitly spelled out, using the same tag numbers. ([-] means the element would have had a string tag in the first approach)

```

AdditionalClientId ::= SEQUENCE {
  id          [0] IMPLICIT IdInfo,
  name        [1] IMPLICIT NameInfo OPTIONAL,
  contact     [2] IMPLICIT ContactInfo OPTIONAL,
  status      [3] IMPLICIT StatusCode -- as above
}

```

```

IdInfo ::= SEQUENCE OF CHOICE { -- this could be a sequence with every member optional
  systemId      [21] ILL-String ,
  social SecurityNo [22] ILL-String,
  accountNo     [23] ILL-String,
  driversLicenceNo [24] ILL-String,
  otherNo       [25] OtherIdInfo
}

```

```

OtherIdInfo ::= SEQUENCE {
  idType      [-] ILL-String,
  idValue     [-] ILL-String
}

```

```

NameInfo ::= SEQUENCE {
  lastName     [31] ILL-String,
  firstName    [32] ILL-String,
  middleName   [33] ILL-String OPTIONAL,
  prefix       [34] ILL-String OPTIONAL,
  suffix       [35] ILL-String OPTIONAL,
}

```

```

ContactInfo ::= SEQUENCE OF CHOICE { -- this could be a sequence with every membe
--optional
  homeAddress   [5] AddressInfo,
  businessAddress [6] AddressInfo,
  otherAddress  [-] OtherAddressInfo,
  emailAddress  [10] EMailInfo
}

```

```

AddressInfo ::= SEQUENCE {

```

```
c/oName          [40] ILL-String,
streetAndNumber  [41] ILL-String,
additionalInfo   [42] ILL-String OPTIONAL, -- e.g. Apt., Floor
boxAddress       [43] ILL-String OPTIONAL,
city-town        [44] ILL-String OPTIONAL,
state-prov-region [45] ILL-String OPTIONAL,
country          [46] ILL-String OPTIONAL,
postalCode       [47] ILL-String OPTIONAL,
voice            [8] TelephoneInfo OPTIONAL,
fax              [7] TelephoneInfo OPTIONAL,
pager           [9] TelephoneInfo OPTIONAL
}
```

```
OtherAddressInfo ::= SEQUENCE {
  addressType      [0] ILL-String,
  addressValue     [1] AddressInfo
}
```

```
EMailInfo ::= SEQUENCE {
  emailService     [60] ILL-String,
  emailAddress     [61] ILL-String
}
```

```
TelephoneInfo ::= SEQUENCE {
  countryCode      [50] ILL-String OPTIONAL,
  areaCode        [51] ILL-String OPTIONAL,
  localNumber     [52] ILL-String,
  extension       [53] ILL-String OPTIONAL
}
```