SIFA UK Object Proposal and Plan:
Curriculum & Timetabling

Submitted by:	Curriculum & Timetabling Task Force
Data Model Version:	1.3
Date Submitted:	1 July 2009
Status:	Draft

Curriculum & Timetabling

Status: **Draft** (objects may change post-trials)

Document Version Control:

Version	*Date*	*Author*	*Notes*
0.1	25/9/08	Chris Johnson	First draft based on meeting of 8/9 Sept 2008 at Bangor.
0.1b	27/9/08	Chris Johnson	Second draft – more objects added.
0.1c	30/9/08	Chris Johnson & John Story	Following John Story's feedback.
0.1d	17/10/08	Chris Johnson	Following task-force discussions at October Quarterly Conference.
0.2	26/11/08	Chris Johnson	Following October telecon, and Andy Ferguson's suggestions.
0.2b	4/12/08	Task Force	Following Task Force meeting.
0.2c	8/12/08	Chris Johnson	Incorporating feedback from December Quarterly Conference.
0.2d	8/3/09	Chris Johnson	Following Andy Ferguson's feedback.
1.0 draft	9/3/09	Task Force	Following feedback from John Story & John Ashworth.
1.1 draft	19/5/09	Chris Johnson & Andy Ferguson	Following feedback from Iain Bullock, Linda Marshall, Andy Ferguson , etc. And following feedback from preparation for initial trials Apr/May 09.
1.2 draft	12/6/09	Chris Johnson & Andy Ferguson	Following further feedback from trials.
1.3 draft	1/7/09	Chris Johnson	Following feedback from Mark Ward.

Table of Contents

Object Proposal

The Business Use Cases for Curriculum and Timetabling

Timetabling in the UK is particularly concerned with scheduling staff, rooms and students to times of the day. US SIF 2.2 objects associated with timetabling refer to daily organisational information and curriculum assessment, which (while essential information) does not cater for information flow that will allow the scheduling of classes or activities.

Currently many schools use a module in their integrated MIS to address the timetabling function. However there are also many schools that use a stand-alone timetabling package, exporting staff, room, subject (and perhaps student) information from the school's MIS into the third party timetabling product. When the timetable is scheduled the information is exported out of the third party product and imported back into the MIS.

Analysis of the current US SIF 2.2 objects/elements provide no real match to the data that MIS need to interface with other third party systems. Therefore a need exists to propose new objects that will enable standardisation of the interaction between packages so that Vendors and applications are able to provide the required interoperability. Improved interoperability between systems will lead to more accurate and consistent data, as well as time savings due to streamlined operations of data transmissions vs. manual exports and imports.

Timetabling is a specialist activity, and a number of schools have begun outsourcing the process of scheduling to external experts. These experts use many of the same applications that would otherwise be used within the school. SIF integration of these applications will allow external experts to securely export data directly from the school, whilst allowing the school to manage exactly what information is released to the timetabler. It would also ensure a secure import of the finished timetable, meaning no more data lost in the post and no mass export just to make sure they get everything.

In the first instance, interaction between third party packages and the MIS is at a school level. However with Federations and School Consortia on the horizon, MISs interacting with each other would be highly desirable. As co-operating schools may not share a common MIS Vendor, interoperability is key.

Four fundamental flows of data have been identified in the timetabling process. Each can be separated into its own business case as a useful independent goal, the implementation of which would make a positive impact on timetabling in a SIF enabled school.

These **four** business cases are described below.

1. Business Use Case for Basic data

A third party timetabling package requires information from which it derives a schedule. Most of this information can be classed as Basic data, arriving in the form of lists of data objects held by another source, usually the MIS. The Basic data may include any of the following:

- People/Staff
  - Teachers
  - Support Staff
  - Posts
- Students (optionally)
- Subjects
- Sites
  - Physical
  - Virtual
- Learning Spaces / Rooms
  - Physical Rooms

    - Virtual Spaces
    - Other Schools

- Teaching Groups
    - Teaching
    - Registration

Many of the lists provided to the timetabler are subject to frequent update and change. The choreography for this business case explains the method of synchronisation employed to handle this in an appropriate manner.

Currently all of this data is typically transferred manually, or relies on individual agreements between suppliers
A push button transfer would improve accuracy and speed, reducing the cost of starting a new timetable. A successful method of synchronisation would reduce the chance that new or updated information was lost before reaching the timetabler, eliminating unnecessary and costly reworking of timetables.

2. Business Use Case for Cycle Structure Data Object

Before a timetable can be constructed the shape of the schedule must be defined. The Cycle Structure includes:

- Number of Days (length of the cycle)
- Number of Periods
- Positions of Breaks
- Positions of Lunches (perhaps staggered)
- Non teaching time
- Registration, etc

MIS systems may make use of cycle information for many purposes. Transfer of a Cycle Structure by SIF will ensure consistency between applications. In the case of a Federation a timetabler may need to be aware of another school's Cycle Structure. Sharing such information would be possible in a shared zone.

3. Business Use Case for Curricular Structure Data Object

Within the constraints of the National Curriculum and available resources, every school sets out an overall curricular structure for its students. In most schools this is described by the Curriculum Diagram. (This diagram does not go as far as defining Learning Aims or Goals, these fall outside the scope of such a document.) A data object for Curricular Structure needs to define the eligibility of student groups according to subject, whilst maintaining a disjoint clash free structure. To achieve this we have arrived at the concept of a 'junction'.

A junction is a labeled logical device that takes one or more student groups as input, transforms them and outputs one or more student groups. The transformation is either an aggregation or a division of the input. A collection of related junctions forms a Curricular Structure. Although it may be possible to construct a Curriculum Diagram from a Curricular Structure, that is not its purpose. No graphical information is transferred, only the relational information. The information held in the Curricular Structure is used by the timetabler as the basis for the schedule and by the MIS for populating a schedule with students using the eligibility rules inherent in the curricular structure. A SIF transfer would be used to ensure that the information held by both systems is consistent.

4. Business Use Case for Schedule Data Object

Once the timetabling process is completed, the schedule (ie: the lessons and the times when they are scheduled) need to be transferred from the timetabling package to the MIS. At the moment this is done either manually, which is subject to human error and time-consuming, or by a proprietary interface, which is subject to cost and the vagaries of data quality. A Vendor agnostic transfer by SIF would reduce the cost and improve the speed of transfer. Working towards interoperability shows up flaws in data, forcing improved standards in all systems.
A published SIF interface will also allow third party suppliers to maintain one generic interface rather than multiple MIS specific interfaces.

There are two ways that a schedule may be considered as a flow of data:

1. The transfer of a completed schedule from the timetabler to the MIS.
2. The delivery of working timetables to users of the schedule. These may in fact be considered as separate business cases. We consider the first type of transfer here.

A schedule transfer involves a package of information. In a header is information telling the recipient about the valid application of the schedule including:

- Start and end dates
- Creation date
- Academic (calendar) year
- Source software

The package of information will always contain the curricular structure that was used to derive the schedule alongside the schedule itself, which is a collection of Lessons/Meetings defined by:

- Subject / Topic / Description
- Teaching Group
- Time / Period
- Teacher / Staff
- Room / Location
- Type (Teaching / Non-teaching)
- etc

Delivery of schedules from the Timetabling application to the MIS needs to be carefully managed. There are three situations where a schedule may be delivered:

1. A trial timetable, not intended for use as a working schedule but for analysis of its potential, it may be incomplete or make use of imaginary staff or resources.
2. Delivery of a complete schedule ready for use.
3. Delivery of a replacement schedule, eg. a major reworking of a timetable would be conducted in the timetabling package and a new schedule delivered to the MIS as substitute for a previous version.

Approval

All four business cases were approved at the SIFA Quarterly Conference in Oct 2008.

Time Line

The proposed objects are scheduled for inclusion into SIF Implementation Specification (UK) 1.2 due for release June 2009. In the meantime, it is intended to apply for “Draft Status” and set-up one or more Pilot Schemes to test the objects.

Use Cases

Use Case 1: TT-Software imports Basic Data

SIF Working Group: Curriculum & Timetabling	**Title:** TT-Software imports Basic Data **Status:** Draft
SIF Version and References	SIF UK Implementation Specification 1.2
Actors	TT-Software (school timetabler), MIS (office staff)
Preconditions	
Summary	TT-Software wants to know what basic data is held by the MIS (teachers,subjects,rooms, etc) as it will use this data in building the timetable
Post Conditions	
Main Sequence of Events	1. TT-Software requests the basic data from MIS 2. MIS receives request and sends the basic data
Alternative Sequence of Events or Variations	
Exceptions	
Open Issues	

Use Case 2: TT-Software sends Timetable Cycle to MIS

SIF Working Group: Curriculum & Timetabling	**Title:** TT-Software sends Timetable Cycle to MIS **Status:** Draft
SIF Version and References	SIF UK Implementation Specification 1.2
Actors	TT-Software (school timetabler), MIS (office staff)
Preconditions	
Summary	TT-Software tells MIS what the cycle is for this timetable (the number of days and periods in the school ‘week’, possibly the position of breaks and lunches, and other related information)
Post Conditions	
Main Sequence of Events	1. TT-Software publishes the cycle 2. MIS receives the cycle
Alternative Sequence of Events or Variations	
Exceptions	
Open Issues	

Use Case 3: TT-Software sends Curriculum Structure to MIS

SIF Working Group: Curriculum & Timetabling	**Title:** TT-Software sends Curriculum Structure to MIS **Status:** Draft
SIF Version and References	SIF UK Implementation Specification 1.2
Actors	TT-Software (school timetabler), MIS (office staff)
Preconditions	
Summary	TT-Software tells MIS what the Curriculum Structure is for this timetable (these are the “rules” used to construct the timetable, including a

	description of how the various Teaching Groups are related, what the student eligibility rules are, etc)
Post Conditions	
Main Sequence of Events	1. TT-Software publishes the curriculum structure 2. MIS receives the curriculum structure
Alternative Sequence of Events or Variations	
Exceptions	
Open Issues	

Use Case 4: TT-Software sends Timetable Schedule to MIS

SIF Working Group: Curriculum & Timetabling	**Title:** TT-Software sends Timetable Schedule to MIS **Status:** Draft
SIF Version and References	SIF UK Implementation Specification 1.2
Actors	TT-Software (school timetabler), MIS (office staff)
Preconditions	
Summary	TT-Software tells MIS what the Timetable Schedule is (the Schedule takes the Teaching Groups described previously in the curriculum structure and schedules them as actual lessons, scheduled at specific times of the week, with specific teachers & rooms. This is what most people think of as "the timetable". It also assigns non-teaching activities to specific times of the week.)
Post Conditions	
Main Sequence of Events	1. TT-Software publishes the Schedule 2. MIS receives the Schedule
Alternative Sequence of Events or Variations	
Exceptions	
Open Issues	

Step-by-Step Guide to effecting each Business Case

The outline steps required to effect each of the 4 business cases, are described below.

1 Case 1: Transfer of Basic Data (only)

Transferring Basic Data only:

Step	TT-Software:	MIS:
1	Requests "Basic Data" from MIS.	MIS receives request and sends "Basic Data" **← Basic Data**

2 Case 2: Transfer of Cycle

Transferring Cycle only:

Step	TT-Software:	MIS:
1	Cycle is published **Cycle →**	

3 Case 3: Transfer of Curricular Structure (only)

This follows the same steps as "Schedule" below, but only the curricular structure (ie the "teaching groups" & "junctions") is included in the message.

This is useful where the MIS wants a list of the expected Teaching Groups and Junctions, but the timetabler has not yet finished the final schedule, so the full Schedule (lesson times, etc) cannot be sent yet.

4 Case 4: Transfer of Schedule (and Curricular Structure)

This is the most useful and common case. The process is:

-- The timetable is scheduled in the TT-Software,
-- When completed, the school transfers the timetable to the MIS.

The steps in this process are as follows:

Transferring the Timetable Schedule:

Step	TT-Software:	MIS:
1	Requests "Basic Data" from MIS (and a list of TGs for next year, if MIS has already set these up)	MIS receives request and sends "Basic Data" **← Basic Data (& Teaching Groups)**
2	The Basic Data is used to run preliminary checks on the teachers, subjects, rooms, etc being used by the two systems. GUIDs (where supplied by the MIS) are added to timetable resources (teachers, subjects, etc) about to be exported, etc.	
3	Timetable is published: **Basic Data →** **Timetable & Curriculum →**	

4		MIS reads the timetable, and its “Gatekeeper” manages the import of data. Note: This timetable replaces any earlier timetables – it does not merely edit or add-to an existing timetable
5		MIS publishes its “Results message”, describing the success (or otherwise) of the import. **← Timetable Results** (NB: format of this file not agreed yet – to be included in next version)
6	Acts on Results Message (if required).	

5 *Notes on the above transfers:*

5.1 Direction of data transfer

The above processes describe the most common use of each case. However the option is left open to transfer data the opposite way, and this may sometimes be useful.

For example, case 2 above describes the TT-Software telling the MIS what the cycle will be for next year’s timetable. However, the MIS could use the same message to tell the TT-Software what the cycle is for this year.

Similar flexibility applies to all the cases.

5.2 Requesting the data

The SIF spec allows us to write queries, so as a refinement to the above, we should also be able to ask "please send me all your WorkforcePersonal who are teachers" or similar, and the system would send the list in a standard way.

5.3 Ownership

It is intended that wherever possible, the MIS will “own” all the data and create its own GUIDs.

The TT-Software refers to all its resources (teachers, subjects, rooms, teaching groups, etc) by its own internal Local Ids, but it includes the GUID it received from the MIS, where known.

So, for example, if the TT-Software sends a Teacher without an MIS GUID, it means this teacher is (probably) known only to the TT-Software, and not the MIS. Whereas, if the TT-Software sends a Teacher with an MIS GUID, it means this teacher already exists in the MIS.

This is described in more detail in later sections.

5.4 Gate-keeper

When receiving information from the TT-Software, the MIS needs to employ a “Gatekeeper” for matching / checking / validating the incoming data and comparing it with its own data.
This is because the data in the MIS is the “master”, and the data arriving from the TT-Software will typically only partly match this – the other part is “proposed” for next year’s timetable (and the MIS needs to decide how to handle this “proposed” data).

5.5 Teachers in Basic Data

When teachers are passed from TT-Software to MIS, they are passed as “TimeTable Teachers”.

When teachers are passed from MIS to TT-Software, they are passed as “WorkforcePersonal“s (or perhaps as “TimeTable Teachers” if the MIS still has them stored that way)

Proposed New Data Objects

The proposed objects are:

(A) Data Objects for **Timetable Transfer**:

A1 Scope

(B) Data Objects for **Basic Data**:

B1 Timetable Teacher
B2 Timetable Subject
B3 Timetable Room
B4 Timetable Site
B5 Timetable Cycle
B6 MIS Subject, Room, Site

(C) Data Objects for **Curriculum & Schedule**:

C1 Teaching Group
C2 Junction (C1 & C2 describe the *Curricular Structure)*
C3 Non-Teaching Activity
C4 Lesson (C4 describes the *Schedule (ie: the lessons & where they are scheduled))*

Current Limitations of these Objects

For clarity, please note that the following are <u>not</u> included in this version of the Data Objects:

1) *Transfer of individual student data*: We have not included this because TT-Software does not necessarily hold a list of individual students; instead, it may work solely in terms of *groups* of students. However, optional objects could be added for TT-Software that does hold such information. *LearnerPersonal* may help here.

2) *Calendar dates of individual lessons*: Again this is not something that TT-Software needs to know. So we have assumed the schedule will work simply on a 'cycle' of days, which may be labeled as simply as "Day1 – Day 10" or "Mon – Fri". They do not have calendar dates associated with them.
Therefore, it is not possible to use these objects to query / report on a student's timetable for a specific date. (The Task Force feels a separate reporting tool could be used for this; to be investigated later).

   These dates could be expanded further to represent carousels, etc. (although the precise dates are very often not the concern of the timetabler).

Conclusion: 1) & 2) above would be useful in some instances and could be added as optional items to the next version.

3) *Further Education*: The Spec does not cover FE. It is clear there are both common areas (teachers, subjects) and differences (eg: in the concept of the "cycle") in FE timetabling, but so far there has been no input from FE experts.

Object Plan

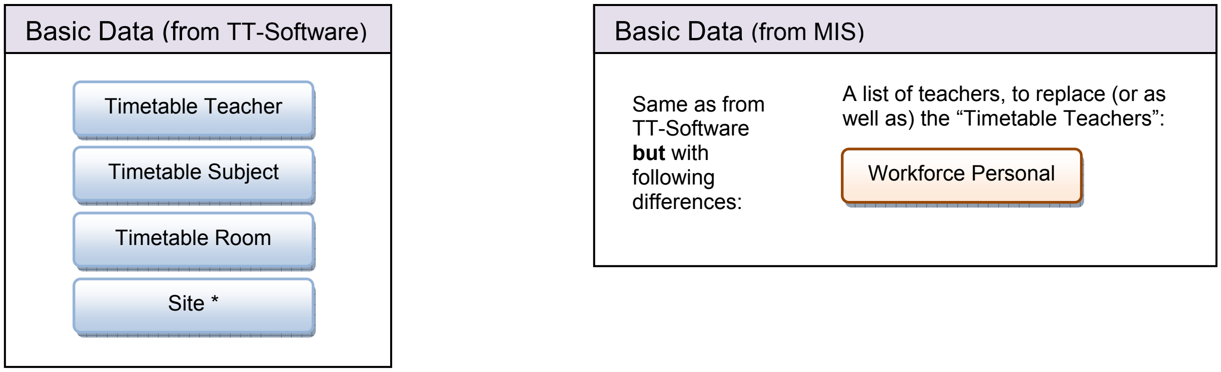
Object Dependencies and Relation Map

The Dependencies and Relationship Map are as follows:

Key to Diagrams:

Workforce Personal	Orange means this object already exists in SIF Spec 1.1
Timetable Teacher	Blue means this is a proposed object (Basic Data)
Timetable Teacher	Green means this is a proposed object (Curriculum Structure)
Room *	Asterisk means this object could be generally useful outside of timetabling too

1 Business Case 1: TT-Software requests Basic Data from MIS (or sends to MIS):



Basic Data (from TT-Software)

- Timetable Teacher
- Timetable Subject
- Timetable Room
- Site *

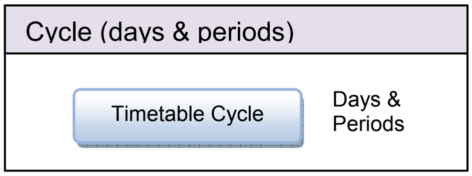
Basic Data (from MIS)

Same as from TT-Software **but** with following differences:

A list of teachers, to replace (or as well as) the "Timetable Teachers":

- Workforce Personal

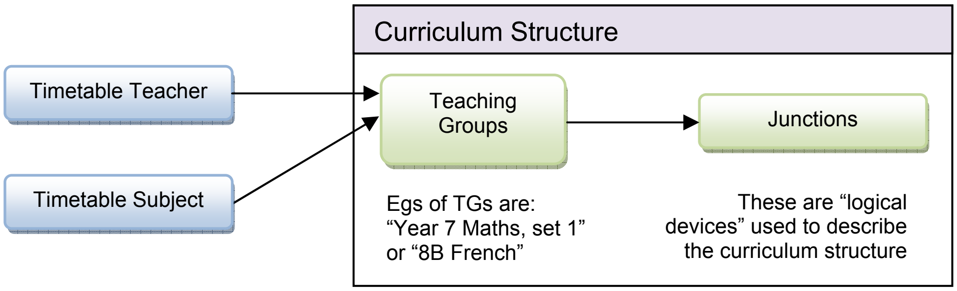
2 Business Case 2: Timetable Cycle:



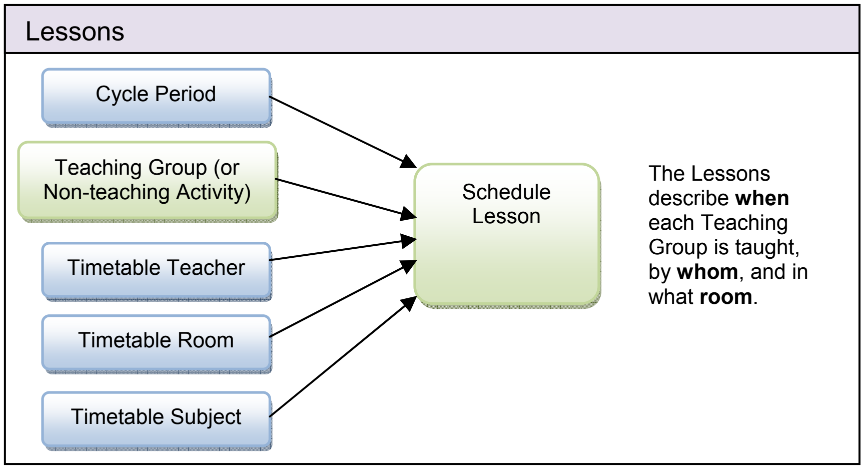
Cycle (days & periods)

Timetable Cycle — Days & Periods

3 Business Case 3: Curricular Structure:


Curriculum Structure
Timetable Teacher
Timetable Subject
Teaching Groups
Junctions
Egs of TGs are: "Year 7 Maths, set 1" or "8B French"
These are "logical devices" used to describe the curriculum structure

4 Business Case 4: Timetable Schedule:


Lessons
Cycle Period
Teaching Group (or Non-teaching Activity)
Timetable Teacher
Timetable Room
Timetable Subject
Schedule Lesson
The Lessons describe **when** each Teaching Group is taught, by **whom**, and in what **room**.

Notes:

1) "**Scope**" should also be included for each transfer.

2) To transfer a **complete timetable**, all the above objects need to be transferred (Basic Data, Cycle, Curriculum Structure and Lessons) , ideally in a single message.

Changes to Existing Objects

None indentified.

Infrastructure Changes

There is not expected to be any impact on the infrastructure.

Glossary of Terms used in Object Definitions

- **MIS**: The **M**anagement **I**nformation **S**ystem used by the school for their Admin, etc.

- **TT-Software**: The Timetabling-software (whether it's a third-party program or a module of the MIS).

- **The timetabler**: The person creating the timetable in the TT-Software.

- **Timetable Teacher**: A teacher on the timetable who may (or may not) exist as a WorkforcePersonal in the MIS.

  -- If s/he does exist in the MIS, TT-Software includes the GUID that the MIS provided when it passed it the Basic Data.

  -- If s/he does not exist in the MIS (so far as we know), then we don't include a GUID.

  The MIS then has to handle this appropriately.

- **Timetable Subject**: same idea as "Timetable Teacher". Ditto for rooms, etc. In fact, for any Basic Data that may differ between TT-Software and the MIS.

- **More:** for a ***general*** list of terms used in timetabling, see the list discussed by the Task Force, at:

  http://www.timetabler.com/Glossary.pdf

Object Definitions

The definitions follow in the next several sections:

Object Definition A1: Timetable Scope

A1.1 Overview

The "Scope" is required whenever a cycle, curriculum structure, or timetable schedule, is transferred.
It tells the receiving system the period of time the timetable covers (dates to/from) as well as various other "general" items of information that help to describe the timetable.

A1.2 Data Object Definition

Scope	Scope of this timetable, ie: a general description of what dates the timetable covers, etc				
	Element	**Char**	**Description**	**Type**	Note
@	RefId	M	Unique GUID for this Timetable transfer	RefIdType	
	TTScopeName	M	A short label for reference in screen displays.	xs:normalizedString maxLength = 50	
	SupplierName	M	The supplier of the software that created this message	xs:normalizedString maxLength = 50	
	SupplierURL	O	A URL to a page on the supplier's website that gives information about the transfer and/or contact details for the supplier. (Alternatively, a telephone number could be given).	xs:normalizedString maxLength = 60	
	Product	M	The product that created this message	xs:normalizedString maxLength = 40	
	Version	M	The version of that Product	xs:normalizedString maxLength = 20	
	SchoolInfoRefId	M	The unique school ID, taken from the existing SchoolInfo object (requested from ZIS)	IdRefId	[1]
	SchoolName	M	The school name in plain text – to help a person to see immediately this is the correct school	xs:normalizedString maxLength = 60	
	ScheduleCreationDate	C	Date schedule was created or last edited (omitted for "Basic data only" message)	xs:date	
	ScheduleName	M	Name given to schedule by timetabler. The TT-Software may contain more than one schedule, so this tells us which one is being sent.	xs:normalizedString maxLength = 20	
	SchoolYear	M	The academic year the timetable or	SchoolYear	

			basic data relates to (given as year the academic year starts in)		
	StartDate	C	First day of the timetable	xs:date	
	EndDate	C	Last day of the timetable	xs:date	
	Status	M	The "Status" of the timetable transfer 'BT' and 'ET' are sent by TT-Software when sending a timetable. The others are sent by the MIS after receiving a timetable. (See also "Reasons for Status" below).	Values: BT = beginning of transfer ET = end of transfer RP = received but import pending SI = successful import UI = unsuccessful import	

A1.3 Notes on Elements:

[1] This means the TT-Software will need to request the SchoolInfoRefId before starting to transfer the timetable.

A1.4 Example XML

```
<Scope RefId="07660F2862CA460BAA95E57AB49CBE88">
  <TTScopeName>Timetable 2008-09</TTScopeName>   for screen displays
  <SupplierName>October Resolutions</SupplierName>
  <SupplierURL>www.timetabler.com</SupplierURL>
  <Product>TimeTabler</Product>
  <Version>4.90.00</Version>
  <SchoolInfoRefId>D6F60F2862CA460BAA95E57AB49CBE88</SchoolInfoRefId>
  <SchoolName>Example High School</SchoolName>   visible name for human checking
  <ScheduleCreationDate>2008-08-25</ScheduleCreationDate>
  <ScheduleName>Final Schedule 08-09</ScheduleName>
  <SchoolYear>2008</SchoolYear>
  <StartDate>2008-09-01</StartDate>
  <EndDate>2009-07-31</EndDate>
  <Status>BT</Status>
</Scope>
```

A1.5 Giving Reasons for the "Status"

If required, the MIS can use **<SIF_ExtendedInfo>** to describe the reasons for the status.

Example: if Status is "UI" (unsuccessful import) then <SIF_ExtendedInfo> might say "Room 23 not recognised".

Object Definition B1: Timetable Teacher (TTTeacher)

B1.1 Overview

The MIS holds a list of current teachers, of type "WorkforcePersonal".
The TT-Software also holds a list of teachers, who will be teaching lessons on the timetable.

However, although similar, these two lists will typically have certain differences. There are several reasons for this, most related to the fact that TT-Software is usually working on next year's timetable, while the MIS holds the list of teachers (and other staff) currently in the school.

Reasons include:

a) the school is intending to hire a new Maths teacher in September, but does not yet know who s/he will be. This requires an extra "Timetable Teacher" (called, say "MA1") in the TT-Software, who does not (yet) exist in the MIS.

b) the timetable wishes to experiment with "what if ?" scenarios; for example, what if the school had an extra French teacher – would it improve the timetable ? This requires an extra "Timetable Teacher" (called, say "FR1") in the TT-Software, who does not exist in the MIS.

c) one of the current Music teachers is leaving this summer, and may not be replaced. So although she continues to exist in the MIS staff list, she will not appear in the list of "Timetable Teachers".

d) the "Teacher" on the timetable actually represents a job-share involving two teachers.

All the above mean that the list of teachers in the TT-Software will typically differ from the list of staff in the MIS. Therefore, the TT-Software works with a new data object called the "**Timetable Teacher**". This is simply a teacher on the timetable. Often (but not always) this "Timetable Teacher" will be one of the "WorkforcePersonalId" people stored in the MIS. Where this is the case, the optional element "WorkforcePersonalId" will contain the "WorkforcePersonalId" recognised by the MIS (see example below). Where the person currently exists only in TT-Software, the "WorkforcePersonalId" is omitted.

In summary, the list of "Timetable Teachers" is the list of teachers (or other staff) used by the TT-Software to teach the lessons on the timetable. It will typically be similar to the "WorkforcePersonal" list in the MIS, but will have some additions and/or deletions.

B1.2 Data Object Definition

Lessons on the timetable are taught by a "Timetable Teacher":

TTTeacher	A teacher (or teaching post) used on the timetable. This may or may not be a "real" teacher stored in the MIS yet.			
	Element	**Char**	**Description**	**Type**
@	RefId	M	Unique GUID for this Timetable Teacher	RefIdType
	Code	M	Unique Code local to TT-Software (could be Teacher Initials, Staff Code, etc)	xs:normalizedString maxLength = 6
	WorkforcePersonalRefId	O	If this teacher is known to the MIS, this is the "WorkforcePersonalId"	IdRefType
	Description	M	This is the description of the teacher (or "post", etc). It may be a name "John Smith" or it may just be a description such as "New Maths Teacher". If WorkforcePersonalId is not supplied, this Description is used by the MIS to help the user manually match to a teacher in the	xs:normalizedString maxLength = 24

			MIS (or decide on some other appropriate action).	

B1.3 Example XML

```
  <TTTeacherList>
This teacher is known to the MIS, and so its WorkforcePersonalId is supplied:
    <TTTeacher RefId=”AEECD89932678725A317B265CE015455”>
      <Code>CJo</Code>
      <WorkforcePersonalRefId>AB34DC0932678725A317B265CE015455</WorkforcePersonalRefId>
      <Description>Chris Johnson</Description>
    </TTTeacher>
This teacher is not known to the MIS, so WorkforcePersonalId is omitted:
    <TTTeacher RefId=”874CD89932678725A317B265CE015455”>
      <Code>MAT</Code>
      <Description>New Maths Teacher</Description>
    </TTTeacher>
...etc for all teachers on the timetable
  </TTTeacherList>
```

B1.4 How the MIS might handle the “Timetable Teacher”

When it imports the timetable, the MIS will have its own internal rules for handling the list of “Timetable Teachers”.

For each imported “Timetable Teacher”, the MIS needs to be able to deal with two possible cases:

Case	Action that could be taken
WorkforcePersonalId has a value	There should be this same WorkforcePersonalId already in the MIS. WorkforcePersonalId should be matched to this WorkforcePersonalId
WorkforcePersonalId omitted	Show the user the “Description” of this Timetable Teacher, and provide a list of all the staff in the MIS, who have not been matched to any other “Timetable Teacher”. Then: (a) The user is invited to match them manually. eg: if the new Maths teacher has now been appointed, the user can manually tell the MIS that “MA1” in the TT-Software is actually “Joanne Jones” in the MIS. (b) If the user cannot match them, then it may be that the new Maths teacher has not been appointed yet. Or there may be another problem. The MIS will have to decide whether to add a temporary/new teacher, or to ignore the imported teacher and his/her lessons, or …? This will require liaising with the timetabler to see what s/he intended by including this unrecognized teacher. (c) The MIS may also choose to hold the “Timetable Teacher” in some form – for eg, as a “temporary post” while it waits for a new member of staff to be appointed. In this way, the “Timetable Teacher” could appear on staff or student timetables in the MIS, even though s/he is not a real teacher yet, and does not appear in WorkforcePersonal

B1.5 Results Message

It is intended in a later version of the objects to include a “Timetable Results” message. Any matching (or other) problems encountered in the MIS during the import, could then be written to this message, which can be read by the TT-Software post-import, to see how successful the import has been, identify problems, etc.

B1.6 Terminology

A Timetable Teacher with a WorkforcePersonalId could be called a “real teacher” (or “existing teacher” or “MIS teacher”). A TimeTable Teacher without a WorkforcePersonalId could be called a “proposed teacher” (or timetable-only teacher).

Object Definition B2: Timetable Subject

B2.1 Overview

Although there is already a list of standard Subjects in the SIF Spec, the list of Subjects used by TT-Software (and possibly by the MIS) will usually be wider than this, to include:
-- variations on a subject (eg: you might have several variants on Maths),
-- projects that are not a specific standard subject
-- “subjects” that actually represent meetings, etc
-- a new subject that doesn’t appear in the standard SIF spec yet

For this reason, we propose the “Timetable Subject”.

As with the “Timetable Teacher”, the “Timetable Subject” may or may not already exist in the MIS.

B2.2 Data Object Definition

All Teaching Groups have a “Timetable Subject”:

TTSubject A subject used on the timetable. This may or may not be a subject stored in the MIS yet.

	Element	Char	Description	Type	Note
@	RefId	M	Unique GUID for this Subject	RefIdType	
	Code	M	Unique Code local to TT-Software (short name)	xs:normalizedString maxLength = 6	
	SubjectRefId	O	If this subject is known to the MIS, this is the MIS’s GUID	IdRefType	[1]
	Description	M	This is the description of the subject (long name)	xs:normalizedString maxLength = 15	

B2.3 Notes on Elements:

[1] What “Subject” object in the MIS should this should refer to ? Is it something in the existing Spec or does it need to be proposed ? See section B6.

B2.4 Example XML

```
  <TTSubjectList>
```

This subject is known to the MIS, and so its MIS GUID is supplied in SubjectRefId:

```
    <TTSubject RefId=”A034DC09334789A317B265CE01765744”>
      <Code>MAT</Code>
      <SubjectRefId>B214DC09334789A317B265CE010A1744</SubjectRefId>
      <Description>Maths</Description>
    </TTTeacher>
```

This subject is not known to the MIS, so SubjectRefId is omitted:

```
    <TTSubject RefId=”A034DC09334789A317B445CE01765701”>
      <Code>CLA</Code>
      <Description>Classical Stud</Description>
    </TTSubject>
```

...etc for all subjects on the timetable

```
  </TTSubjectList>
```

B2.5 How the MIS might handle the "Timetable Subject"

When it imports the timetable, the MIS needs to look for any TimeTable Subject that does not have a SubjectRefId, and handle it appropriately.
TimeTable Subjects that do have a SubjectRefId should be matched to that subject in the MIS.

Object Definition B3: Timetable Room

B3.1 Overview

We can't see a list of "rooms" or "locations" or "learning spaces" in the SIF Spec ?

We therefore propose the "Timetable Room".

The list of rooms being used by the TT-Software may differ from the list of current rooms stored by the MIS.

Reasons for this include:
-- the school is intending to build a new laboratory next year.
-- the school is linking with other schools or local businesses next year, and needs to represent them as "rooms" on the timetable
-- next year's timetable requires some "virtual" rooms

Note: a "room" in this sense does not have to be a physical room; it can be any "learning space" (physical or virtual) including other schools or businesses. See also the section on "Sites".

B3.2 Data Object Definition

Most (but not necessarily all) lessons on the timetable have a "Timetable Room":

TTRoom — A room (or learning space) used on the timetable. This may or may not be a "real" room stored in the MIS yet.

	Element	Char	Description	Type	Note
@	RefId	M	Unique GUID for this Room	RefIdType	
	Code	M	Unique Code local to TT-Software (short name)	xs:normalizedString maxLength = 5	
	RoomRefId	O	If this room is known to the MIS, this is the MIS's GUID	IdRefType	[1]
	Description	M	This is the description of the room (long name) It is used by the MIS to help the user manually match to a room in the MIS (or decide on some other appropriate action).	xs:normalizedString maxLength = 10	
	SiteRefId	O	Which "site" this room is on. It may also be used by the MIS.	IdRefType	

B3.3 Notes on Elements:

[1] What "Room" object in the MIS should this should refer to ? Is it something in the existing Spec or does it need to be proposed ? See section B6.

B3.4 Example XML

```
  <TTRoomList>
```

This room is known to the MIS, and so its MIS GUID is supplied in RoomRefId:

```
    <TTRoom RefId="6C34DC09334789A317B265CE01765744">
      <Code>12</Code>
      <RoomRefId>A034DC09334789A317B265CE01765744</RoomRefId>
      <Description>Room 12</Description>
    </TTRoom>
```

This room is not known to the MIS, so RoomRefId is omitted:

```
    <TTRoom RefId="6C34DC09334789A317B265CE01765744">
      <Code>Lab3</Code>
      <Description>3rd Lab</Description>
      <SiteRefId>1</SiteRefId>
    </TTRoom>
```

...etc for all rooms on the timetable

```
  </TTRoomList>
```

B3.5 How the MIS could handle the "Timetable Room"

When it imports the timetable, the MIS needs to look for any TimeTable Room that does not have a RoomRefId, and handle it appropriately.
TimeTable Rooms that do have a RoomRefId should be matched to that room in the MIS.

Object Definition B4: Timetable Site

B4.1 Overview

There doesn't appear to already be a "site" object in the SIF Spec ?

We therefore propose the "Timetable Site".

The list of sites being used by the TT-Software may differ from the list of current sites stored by the MIS.

Reasons for this include:
-- the school is intending to move to split-site next year.
-- the school is linking with other schools or local businesses next year, and needs to represent them as "sites" on the timetable
-- next year's timetable requires some "virtual" sites to represent some of its lessons

Note: a "site" in this sense does not have to be a physical location; it can be any "learning area" or collection of "rooms" (physical or virtual) including other schools or businesses. Neither does the use of "sites" mean that a school is physically on 2 or more sites, as a single-site school could be divided into 2 or more "timetable sites" in the eyes of the timetabler.

B4.2 Data Object Definition

Rooms (optionally) have a "Timetable Site":

TTSite A site used on the timetable. This may or may not be a "real" site stored in the MIS yet.					
	Element	**Char**	**Description**	**Type**	Note
@	RefId	M	Unique GUID for this Site	RefIdType	
	Code	M	Unique Code local to TT-Software	xs:normalizedString maxLength = 5	
	SiteRefId	O	If this site is known to the MIS, this is the MIS's GUID	IdRefType	[1]
	Description	M	This is the description of the site. It is used by the MIS to help the user manually match to a site in the MIS (or decide on some other appropriate action).	xs:normalizedString maxLength = 10	
	Type	O	What "type" of site this is (if known). It is used by the MIS to help with any manual matching.	Values: S = a site on your school V = virtual (ie: not a physical site) E = an external site (eg: another school in a federation, or a local business)	

B4.3 Notes on Elements:

[1] What "Site" object in the MIS should this should refer to ? Is it something in the existing Spec or does it need to be proposed ? See section B6.

B4.4 Example XML

```
<TTSiteList>
  <TTSite RefId="AF57DC09701928365AA0176890234654">
    <Code>1</Code>
    <Description>Upper Sch</Description>
  </TTSite>

...etc for all sites
</TTSiteList>
```

B4.5 How the MIS handles the "Timetable Site"

When it imports the timetable, the MIS needs to be able to look for any TimeTable Site that does not have an SiteRefId, and handle it appropriately.
TimeTable Sites that do have a SiteRefId should be matched to that site in the MIS.

B4.6 Importance of "Site"

Knowing which site a room is on is important for:

-- scheduling, as teachers and students may not be able to commute quickly between sites
-- cover for absent staff
-- locating the whereabouts of a student or teacher
-- knowing when a student is away from the school (eg: at a local college or local business).

A timetable for a single-site school may not have the concept of a "site", so if this object is omitted from the message, it means everything is conducted on one site.

Object Definition B5: Timetable Cycle

B5.1 Overview

The “Cycle” is a description of the days and periods of the timetable cycle. It represents the period of time before the timetable repeats itself. A cycle is typically a week or fortnight, but could be any period of time.

B5.2 “Rectangular Grid” Representation of the Cycle

The “Cycle” is represented here by a “rectangular grid” of “CycleDays” by “PeriodsPerDay”. This represents a rectangular school week in a quick and obvious way, but it is also useful when drawing a non-rectangular timetable in reports, etc. See next item for details.

B5.3 Non-rectangular weeks

If the school week is **non**-rectangular , then the “unused” periods are given a type ‘X’. **A quick example** demonstrates the point: a school has 6 periods Mon-Thu but Fri is a shorter say with only 5 periods. But this information does not in itself tell us whether the students start late on Fri, or finish early. (Period start & finish times would tell us, but these are optional). We get round this by representing the week as a rectangular grid of 5 days by 6 periods, then looking to see which period on Fri is given type ‘X’ (period 1 or 6).

B5.4 Breaks, Lunches, Registration, etc

In practice, “PeriodsPerDay” will usually also include breaks and lunches, which are given a type ‘B’ or ‘L’, for eg:

Period:	1	2	3	4	5	6	7
Mon	T	T	B	T	L	T	T
Tues	T	T	B	T	L	T	T
Wed	T	T	B	T	L	T	T
Thu	T	T	B	T	L	T	T
Fri	T	T	B	T	L	T	X
Sat	T	T	X	X	X	X	X

This school has a short Fri, then works Sat morning (ie: non-rectangular) and has included periods for ‘B’reak and ‘L’unch

B5.5 “Standard Timetable Day”

There is also the concept of the “**standard timetable day**” which is a typical / default day. In a non-rectangular cycle, any differences from the “standard timetable day” can be specified within the “Cycle /TTDayList /TTDay /PeriodList /Period” structure. So in the above example, the “standard timetable day” would be like Mon, then last period Fri would be specified as an exception, as would periods 3-7 on Sat.

B5.6 Data Object Definition

The cycle comprises a data object “Cycle” with a Common Element “Period”.

The Period (*common element*):

Period A period in the cycle (*Common Element*)					
	Element	**Char**	**Description**	**Type**	Note
	PeriodIndex	M	Ordinal position of period within the day (1,2,3,etc)	integer	
	ShortName	C	A short period label	xs:normalizedString maxLength = 5	[1]
	LongName	C	A longer period label	xs:normalizedString maxLength = 25	[1]

	StartTime	O	Time the period starts	xs:time	[2]
	EndTime	O	Time the period ends	xs:time	[2]
	Type	M	What this period represents. Note that currently some programs represent registration, breaks and lunches as a “period” and some don’t. This Type could differ from the lesson Type (see section C4) if this teacher is doing something in this period which differs from the majority of the school.	Value: T = teaching period R = registration B = break L = lunch O = outside school hours (eg: clubs) N = other non-teaching X = ‘unused’ period (on a non-rectangular cycle)	

The Cycle:

Cycle	The timetable cycle			
	Element	**Char**	**Description**	**Type**
@	RefId	M	Unique GUID for this cycle	RefIdType
	CycleName	O	A label for this cycle	xs:normalizedString maxLength = 40
	CycleDays	M	Number of days in cycle	Integer
	PeriodsPerDay	M	(Max) Number of periods per day (see “Rectangular Grid Representation above)	Integer
	StandardPeriodList	M		List
	StandardPeriodList/Period	MR	Periods in a “standard timetable day” (common element Period)	Period
	TTDayList	M	List all the days of the cycle	List
	TTDayList/TTDay	MR	The days in the cycle	TTDay
	TTDayList/TTDay/TTDayId	M	A unique local ID for this day	xs:normalizedString maxLength = 5
	TTDayList/TTDay/ShortName	M	A short day label	xs:normalizedString maxLength = 15
	TTDayList/TTDay/LongName	M	A longer day label	xs:normalizedString maxLength = 25
	TTDayList/TTDay/DayIndex	M	The day’s position in the cycle (1,2,3,etc).	Integer
	TTDayList/TTDay/PeriodList	O	List of the periods on this day – only need to list the periods that are different from the “standard timetable day” (if any) – any period not listed here is taken to be the same as in the “standard timetable day”	List
	TTDayList/TTDay/PeriodList /Period	MR	A period (see common element Period)	Period

B5.7 Notes on Elements:

[1] These elements are *Mandatory* when **Period** appears in **StandardPeriodList**.

[2] Within any given day, the “StartTime” of a period shall not be earlier than the “EndTime” of any earlier period. Where the “StartTime” of a period differs from the “EndTime” of the previous period, this signifies a ‘hidden’ break or lunch (or similar) for which a period has not explicitly been included on the timetable.

B5.8 Example XML

For a **10-day fortnightly** cycle of **5 (teaching) periods per day**, with **registration** added as an extra non-teaching period at the start of each day, and **differences** (on some days) from the standard day:

```
<Cycle RefId=“07660F2862CA460BAA95E57AB49CBE88“>
    <CycleName>Cycle for Final Schedule</CycleName>
    <CycleDays>10</CycleDays>
    <PeriodsPerDay>5</PeriodsPerDay>    this cycle is based on a grid of “10 days by 5 periods”
    <StandardPeriodList>
        <Period>
          <PeriodIndex>1</PeriodIndex>
          <ShortName>Reg</ShortName>
          <LongName>Registration</LongName>
          <StartTime>08:45</StartTime>
          <EndTime>08:50</EndTime>
          <Type>R</Type>
        </Period>
        <Period>
          <PeriodIndex>2</PeriodIndex>
          <ShortName>2</ShortName>
          <LongName>Period 1</LongName>
          <StartTime>08:50</StartTime>
          <EndTime>09:50</EndTime>
          <Type>T</Type>
        </Period>
        <Period>
          <PeriodIndex>3</PeriodIndex>
          <ShortName>3</ShortName>
          <LongName>Period 2</LongName>
          <StartTime>09:50</StartTime>
          <EndTime>10:50</EndTime>
          <Type>T</Type>
        </Period>
Etc for other periods on ‘standard day’
    </StandardPeriodList>

    <TTDayList>
     <TTDay>
      <TTDayId>DY01</TTDayId>
      <ShortName>Mon A</ShortName>
      <LongName>Monday Week A</LongName>
      <DayIndex>1</DayIndex>
“Mon” has no <PeriodList> as it follows the “standard timetable day”
     </TTDay>

     <TTDay>
      <TTDayId>DY02</TTDayId>
      <ShortName>Tue A</ShortName>
      <LongName>Tuesday Week A</LongName>
      <DayIndex>2</DayIndex>
      <PeriodList>
 “Tue” differs from the “standard timetable day” in periods 2 & 3 (it’s start / end times are different)
        <Period>
          <PeriodIndex>2</PeriodIndex>
          <StartTime>08:50</StartTime>
          <EndTime>09:40</EndTime>
          <Type>T</Type>
        </Period>
        <Period>
          <PeriodIndex>3</PeriodIndex>
          <StartTime>09:40</StartTime>
          <EndTime>10:50</EndTime>
          <Type>T</Type>
        </Period>
      </PeriodList>
     </TTDay>
```

Similarly for Wed A, Thu A, Fri A, Mon B, Tue B, Wed B, Thu B

```
     <TTDay>
      <TTDayId>DY10</TTDayId>
      <ShortName>Fri B</ShortName>
      <LongName>Friday Week B</LongName>
      <DayIndex>10</DayIndex>
      <PeriodList>
```

“Fri B” differs from the “standard timetable day” in that there is no period 5 (day finishes early)

```
        <Period>
          <PeriodIndex>5</PeriodIndex>
          <Type>X</Type>    “unused” period
        </Period>
      </PeriodList>
     </TTDay>
    </TTDayList>

</Cycle>
```

B5.9 Further discussion on Breaks, Lunches, Registration, etc

Currently there doesn’t seem to be a consistent treatment of these – it varies between timetabling packages, between MIS, and may even vary between schools using the same software.
Non-teaching periods such as “lunch”, “break”, “registration”, “after-school clubs”, etc may be treated as a “period” in the timetable, or may not. Often the choice is left to the school, and the timetabler may treat one or more of these as periods at his discretion, omitting the others from the timetable.

One good reason for adding them as a period is if they need covering when staff are absent, although even there the choice may depend on how the school’s “Cover software” works.

This doesn’t matter provided the MIS allows breaks/lunches/registrations to be added post-transfer.

B5.10 “Type” can be over-ridden by individual lessons

An individual lesson can over-ride the “Type” of the period. For example, a 6th-form lesson is scheduled during the lunch-hour. The lesson has type ‘T’ even though the period it is scheduled into has type ‘L’.

B5.11 Staggered lunches

An additional complication is that a school may have “staggered” lunches.

If we used the above method to represent lunch, then we could do the following to cope with 'staggered' lunches for different year-groups (or individual registration groups):

In the cycle: add two periods for lunch (“Lunch 1” and “Lunch 2”).

In the schedule: - for the students taking an **early lunch**, schedule non-teaching lessons to their registration groups (or years) in “Lunch 1” (leaving those groups free in “Lunch 2”).

- for the students taking the **late lunch**, reverse the process: schedule non-teaching lessons to their registration groups (or years) in “Lunch 2” (leaving those groups free in “Lunch 1”).

NB: Breaks and registration may also vary for different years, but they could be handled the same way.

Example XML: for a day with registration, staggered lunches and breaks, can be supplied on request.

Object Definitions B6: “MIS” Subject, Room, Site

B6.1 Overview

We can’t see the following lists in the current SIF Spec ?:

-- a list of subjects as they would be used on a timetable
-- a list of “rooms” (or “locations” or “learning spaces”)
-- a list of “sites”

At some stage, we may therefore need to propose these objects, so that these concepts can be stored in the MIS.

B6.2 Possible Data Object Definitions

Taking the Room object as an example, it would be similar to TTRoom, as follows:

Possible **Data Object Definition for “Room”:**

Room	A room (or learning space) in the school ***Possible Definition***			
	Element	**Char**	**Description**	**Type**
@	RefId	M	Unique GUID for this Room	RefIdType
	Code	M	Unique Code (short name)	xs:normalizedString maxLength = 5
	Description	M	This is the description of the room (long name)	xs:normalizedString maxLength = 10
	SiteRefId	O	Which “site” this room is on (see `TTSite`)	IdRefType

B6.3 Suggested way forward

Until (MIS) Room is defined, we cannot transfer from MIS → TT-Software as there is no mechanism to transfer (MIS) rooms, etc. In the meantime, the following is proposed:

- For now, the initial scope is for a one way transfer (TT-Software → MIS)
- We keep the optional reference to the (MIS) Room as this will be required once an (MIS) Room has been defined
- There will be a future update to this Spec where we can define (MIS) Room etc, once we’ve got the one way transfer working properly.

Object Definition C1: Teaching Group

C1.1 Overview

A “Teaching Group” is defined as “a group of students brought together for the teaching of a specific subject”. (They are called “Classes” or “Teaching Sets” in some MIS).

Examples of Teaching Groups (TG) are:
-- 7A French
-- 7B French
-- Year 7 Maths Set 1
-- Year 7 Maths Set 2
-- etc

Each TG will be scheduled for one or more lessons on the timetable.

[Quick Aside: At the moment we are just looking at a simple list of the Teaching Groups – for a description of how they fit into the curricular structure; see the section on ***Junctions***].

C1.2 Data Object Definition

A “Teaching Group” looks like this:

TeachingGroup	A Teaching Group (as defined above)			
	Element	**Char**	**Description**	**Type**
@	RefId	M	Unique GUID for this TG	RefIdType
	MISTGRefId	O	If this TG is known to the MIS, this is the MIS’s GUID	IdRefType
	GroupDefinition	C	*Required* if MISTGRefId omitted (ie: if this is a new (proposed) TG) *Optional* if MISTGRefId included	GroupDefinition (see below)
	MainTeacherRefId	O	The main teacher / supervisor / etc for this group (for reporting / recognition purposes) (see TTTeacher)	IdRefType
	GroupTeacherList	O	A list of any other teachers associated with this TG	List
	GroupTeacherList/ GroupTeacher	MR	A teacher associated with this TG	
	GroupTeacherList/ GroupTeacher /TTTeacherRefId	M	The teacher (see TTTeacher)	IdRefType

	GroupTeacherList/ GroupTeacher /Type	O	The teacher’s relationship with the TG	Values:
				A = Assistant
				H = Head-of-Dept / Subject-Leader
				Z = other / unspecified

GroupDefinition element is defined as follows:

GroupDefinition	Definition of a new Teaching Group				
	Element	**Char**	**Description**	**Type**	Note
	ShortName	M	A short free-format label that describes the group. Used for display purposes in the MIS or TT-Software.	xs:normalizedString maxLength = 12	
	LongName	M	A longer label to help office staff understand the meaning of the group	xs:normalizedString maxLength = 80	
	TTSubjectRefId	M	Subject being taught - see TTSubject NB: We do not allow a TG to have no subject	IdRefType	
	Set	O	Set number (typically 1,2,3,etc or A,B,C, etc but could be a longer label)	xs:normalizedString maxLength = 5	[1]
	Block	O	Block number or letter (eg: "A")	xs:normalizedString maxLength = 5	[1]
	Level	O	Level / Assessment Stage, eg "KS3" (if known / relevant)	Assessment Stage	

C1.3 Notes on Elements:

[1] "Set" and "Block" are optional identifiers that can give more information about the TG . They can be useful when office staff are trying to "recognise" a TG after it has been transferred to the MIS from the TT-Software, or if the MIS stores such parameters (as some do).
Example 1: if you have 5 Maths sets, it is useful to give them sets of 1,2,3,4,5.
Example 2: If you have French in Block A and also in Block C, if is useful to give them "Block IDs" of A or C.

C1.4 Example XML

The XML below is based on Example 4 in the document "***SIFA-CurriculumDiagrams-JunctionDiagrams.pdf***" on the Curriculum & Timetabling Discussion Forum:

ie: Curriculum Diagram for this example is this:

7A	En	Ma	Sc	Fr	Hi	Gg	Ar	Mu	RE	SE	Te	PE
											Te	
7B	En	Ma	Sc	Fr	Hi	Gg	Ar	Mu	RE	SE	Te	PE
7C	En	Ma	Sc	Fr	Hi	Gg	Ar	Mu	RE	SE	Te	PE
											Te	
7D	En	Ma	Sc	Fr	Hi	Gg	Ar	Mu	RE	SE	Te	PE

Just a simple list of the TeachingGroups at this stage – Junctions to follow:

```
<TeachingGroupList>
```

3 Technology groups in upper-half-year:

TG is not known to MIS:

```
<TeachingGroup RefId="AEECD89932678725A317B265CE015455">
```

```
    <GroupDefinition>
      <ShortName>7P/Te1</ShortName>
      <LongName>7P Technology Set1</LongName>
      <TTSubjectRefId>1552098761827530018AB42367CFF301</TTSubjectRefId>     (Technology)
      <Set>1</Set>
      <Block>P</Block>
    </GroupDefinition>
  </TeachingGroup>
```

TG is not known to MIS:

```
  <TeachingGroup RefId=”AEECD89932678725A317B265CE015456”>
    <GroupDefinition>
      <ShortName>7P/Te2</ShortName>
      <LongName>7P Technology Set2</LongName>
      <TTSubjectRefId>1552098761827530018AB42367CFF301</TTSubjectRefId>     (Technology)
      <Set>1</Set>
      <Block>P</Block>
    </GroupDefinition>
  </TeachingGroup>
```

Similarly for 3rd Technology group

… and for PE groups across the year

… and for all other TeachingGroups (7A/En, 7A/Ma, 7A/Sc, etc)

TG is known to MIS:

```
  <TeachingGroup RefId=”AEECD89932678725A317B265CE015441”>
    <MISTGRefId>D6F60F2CA460BAA95E57AB49CBE88</MISTGRefId>          (TG known to the MIS)
  </TeachingGroup>

</TeachingGroupList>
```

C1.5 Importance of requesting a specific School Year

If the TT-Software requests a list of TGs from the MIS, then (i) the request must be careful to specify which **school year** it wants, and (ii) if next year is requested, then the MIS will want to first promote TGs from this year into next year.

Object Definition C2: Junction

C2.1 Overview

“Junctions” are different from the other objects in this Spec, as they do not describe a real thing in school (no timetabler talks about “junctions”). Instead it is a “behind the scenes” device to describe the “**curriculum structure**” behind the timetable. The timetabler will draw a “Curriculum Diagram” (on paper or in a spreadsheet) before starting the timetable; it describes the plan s/he is working to, and describes the student eligibility rules. Junctions are simply our way of representing that diagram in XML.

The curricular structure can be thought of as a “hierarchy” or “tree”, with “School” at the top, and “Teaching Group” at the end of each branch. It can be described by a collection of “junctions”.

A junction is a logical device that inputs one or more student groups, combines or divides them, then outputs them as one or more new student groups. It is best understood by looking at specific examples (see the document “***SIFA-CurriculumDiagrams-JunctionDiagrams.pdf***” on the Curriculum & Timetabling Discussion Forum).

The output(s) from one junction typically form the input(s) for other junctions.
A collection of related junctions forms a curricular structure.

This curricular structure should describe the student eligibility rules inherent in the curriculum.

There are 3 types of input:

Type	Meaning
SC	Represents the “school” … ie: the top of the “tree”
YR	See explanation below
GR	Every other input group

There are 3 types of output:

Type	Meaning
YR	A school year (eg: “Year 7”) NB: although typically a year, in fact “YR” can represent any “*principal division*” of the school, for eg: a combination of years, or in theory, any other population of students.
TG	The "end" of any branch of the "junction tree" is a TeachingGroup … ie: a real group of students being taught a subject (eg: "7P/Te1"). TGs are described in more detail in their own section.
GR	Any other output group required to represent the curriculum diagram (eg: “Band P”).

For more discussion of the use of junctions, and examples, see the document “***SIFA-CurriculumDiagrams-JunctionDiagrams.pdf***” on the Curriculum & Timetabling Discussion Forum.

C2.2 Data Object Definition

Junction A representation of the Curriculum Diagram (as described above)					
	Element	**Char**	**Description**	**Type**	Note
	Label	M	A free-format label that describes the junction. Used for display purposes in the MIS and/or TT-Software.	xs:normalizedString maxLength = 40	
	ShortLabel	M	A shorter version of Label, for display purposes	xs:normalizedString maxLength = 20	
	LocalType	O	An optional way for the TT-Software to provide information about the “type” of junction (eg: “band” or “block”).	xs:normalizedString maxLength = 30	[1]

	InputList	M	List of one or more input groups	List	
	InputList/Input	MR	The input(s) into this junction		
	InputList/Input/**Type**	C	Conditional and has only one value: “SC” “SC” has special meaning – it means this is the top of the “tree” and the input is the whole school	Values: SC = School	
	InputList/Input/ **GroupLocalId**	C	If Type != “SC”, this is the LocalId of another junction output	xs:normalizedString maxLength = 12	
	OutputList	M	List of one or more output junctions	List	
	OutputList/Output	MR	The input(s) into this junction		
	OutputList/Output/**Type**	M	“TG” has special meaning – it means this is the bottom of the “tree” and the output is a TeachingGroup. “YR” means a year (or “principal division”) within the school (7,8,9,etc).	Values: TG = TeachingGroup YR = Year GR = other Groups *Could be expanded if need arises*	
	OutputList/Output/ **LocalType**	O	An optional way for the TT-Software to provide information about the “type” of junction (eg: “band” or “block”).	xs:normalizedString maxLength = 30	
	OutputList/Output/ **GroupLocalId**	C	Unique LocalID of this junction output (required if “Output/Type” is <u>not</u> “TG”)	xs:normalizedString maxLength = 12	[2]
	OutputList /Output/ **TeachingGroupRefId**	C	If Type = “TG” then include here the TeachingGroupID (see TeachingGroup)	IdRefType	
	OutputList /Output/**Desc**	C	Description. Required if Type is <u>not</u> “TG”. Optional if Type <u>is</u> “TG” (since a TG has its own < ShortName> & <LongName> already)	xs:normalizedString maxLength = 40	
	OutputList /Output/ **ShortDesc**	C	A shorter version of Desc, for display purposes	xs:normalizedString maxLength = 12	
	OutputList/Output/ **NCYearList**	C	Can only be present if ‘Output/Type’ = “YR” and even then, it is optional. Used to denote the NC Year Group of the “YR” (or give more than one NC Year Group if this is a combined-year population)	List	[3]
	OutputList/Output/ NCYearList/**NCYear**	MR	NC Year Group	NCYearGroup	

C2.3 Notes on Elements:

[1] “Type” can be used to provide a “behaviour expectation” to the MIS. Eg: all junctions of type “Block” will typically have similar behaviour.

[2] “GroupLocalId**”** is only used if this output forms the **input** for another junction; therefore it is <u>not</u> necessary if the Output Type = “TG”. Another way of thinking of this is that each output will have <u>either</u> a “GroupLocalId” <u>or</u> a “TeachingGroupRefId”.

[3] A school will often be divided into NCYears (eg: 7,8,9,10,11,12,13) but it does not have to be. NCYears may not be used at this school, or two or more years may be combined (eg: 12+13).

C2.4 Example XML

Here are some common examples of how to use junctions:

```
<JunctionList>
```

This junction splits the whole school into its years or 'principal divisions':

```
 <Junction>
  <Label>School years</Label>
  <ShortLabel>Years</ShortLabel>
  <LocalType>Main Divisions</LocalType>

  <InputList>
    <Input>
      <Type>SC</Type>
    </Input>
  </InputList>

  <OutputList>
    <Output>
      <GroupLocalId>Year7</GroupLocalId>
      <Type>YR</Type>
      <Desc>Year 7</Desc>
      <NCYearList>
        <NCYear>7</NCYear>
      </NCYearList>
    </Output>
    <Output>
      <GroupLocalId>Year8</GroupLocalId>
      <Type>YR</Type>
      <Desc>Year 8</Desc>
      <NCYearList>
        <NCYear>8</NCYear>
      </NCYearList>
    </Output>
```

Etc for all years

```
  </OutputList>
 </Junction>
```

This junction describes how this Year 7 divides into its basic groups; 7A,7B,7C (often reg./tutor-groups/forms)

```
      <Junction>
        <Label>Basic Groups in Year 7</Label>
        <ShortLabel>Year7 Groups</ShortLabel>
        <LocalType>Basic Groups</LocalType>

        <InputList>
          <Input>
            <GroupLocalId>Year7</GroupLocalId>
          </Input>
        </InputList>

        <OutputList>
          <Output>
            <GroupLocalId>C#021/7A</GroupLocalId>
            <Type>GR</Type>
            <LocalType>Group</LocalType>
            <Desc>7A</Desc>
            <ShortDesc>7A</ShortDesc>
          </Output>
          <Output>
            <GroupLocalId>C#022/7B</GroupLocalId>
            <Type>GR</Type>
            <LocalType>Group</LocalType>
            <Desc>7B</Desc>
            <ShortDesc>7B</ShortDesc>
          </Output>
          <Output>
            <GroupLocalId>C#024/7C</GroupLocalId>
            <Type>GR</Type>
            <LocalType>Group</LocalType>
            <Desc>7C</Desc>
            <ShortDesc>7C</ShortDesc>
          </Output>
        </OutputList>
      </Junction>
```

This junction describes one of the ways in which the basic groups defined earlier combine into 'bands' (eg: half-year bands):

```
      <Junction>
        <Label>7AB (p)</Label>
        <ShortLabel>7p</ShortLabel>
        <LocalType>Band</LocalType>

        <InputList>
          <Input>
            <GroupLocalId>C#021/7A</GroupLocalId>
          </Input>
          <Input>
            <GroupLocalId>C#022/7B</GroupLocalId>
          </Input>
        </InputList>

        <OutputList>
          <Output>
            <GroupLocalId>C#023/7AB</GroupLocalId>
            <Type>GR</Type>
            <LocalType>Band</LocalType>
            <Desc>7AB (p)</Desc>
            <ShortDesc>7p</ShortDesc>
          </Output>
        </OutputList>
      </Junction>
```

This junction describes how the band defined above might be taught in 2 Maths sets (called 7p/Ma1 and 7p/Ma2):

```
      <Junction>
        <Label>7p Ma</Label>
        <ShortLabel>7p Ma</ShortLabel>
        <LocalType>Block</LocalType>

        <InputList>
          <Input>
            <GroupLocalId>C#023/7AB</GroupLocalId>
          </Input>
        </InputList>

        <OutputList>
          <Output>
           <Type>TG</Type>
           <LocalType>TeachingGroup</LocalType>
           <TeachingGroupRefId>000000000000000002437702F4D6131</TeachingGroupRefId>
           <Desc>7p/Ma1</Desc>
           <ShortDesc>7p/Ma1</ShortDesc>
          </Output>
          <Output>
           <Type>TG</Type>
           <LocalType>TeachingGroup</LocalType>
           <TeachingGroupRefId>000000000000000002437702F4D6132</TeachingGroupRefId>
           <Desc>7p/Ma2</Desc>
           <ShortDesc>7p/Ma2</ShortDesc>
          </Output>
        </OutputList>
      </Junction>
```

Etc for all junctions

```
</JunctionList>
```

C2.5 Junction Rules and Relationships

Each junction will take one of the following forms, which are largely determined by the type and number of <Input>s and <Output>s that they contain:

‘School’ Junction:

- This represents a starting point for dividing all of the students in the school.
- There should be **exactly one** of these junctions.
- It is characterised by having:
  - Exactly one <Input> with <Type>SC</Type>   (SC = ‘School’)
  - One or more <Output>s all with <Type>YR</Type>   (YR = ‘Year’)

‘Year’ Junction:

- This represents the initial divisions that are used for a ‘YR’ that has been output from the ‘School’ Junction.
- There should be **exactly one** of these junctions for each ‘YR’ that is output by the ‘School’ Junction.
- This sort of junction is characterised by having:
  - Exactly one <Input> that links to a ‘YR’ that was defined by the ‘School’ junction.
  - One or more <Output>s all with <Type>GR</Type> (GR = ‘Group’) or (more rarely) <Type>TG</Type> (TG = ‘Teaching Group’)
  - Typically, there would only be a single Junction for each YR that would be used to create ‘Basic Groups’ or ‘Base Bands’ (or similar) for that year. These groups could then be combined using ‘General’ Junctions to produce GR or TG type outputs.

‘General’ Junction:

- Remaining junctions will combine and distribute groups into other Groups and or Teaching Groups
- There may be many of these junctions that will combine together to form a ‘tree’ type network with ‘TG’ (Teaching Group) type outputs as the ‘Leaves’
- This sort of junction is characterised by having:
  - One or more <Input>s that link to a ‘GR’ (group) that was defined as an <Output> from another <Junction>. These <Input>s will not be linked to items that are of <Type> ‘YR’ or ‘TG’.
  - One or more <Output>s; each of these will be either:
    - <Type>GR</Type>   (GR = ‘Group’)

    OR
    - <Type>TG</Type>   (TG = ‘TeachingGroup’).

C2.6 Order of Junctions

Ideally, to assist readability, the 'SC' junction should appear first, with other junctions ‘in order’ - ie: for any given junction, each of its **inputs** should have been identified as the **output** of a junction that appeared earlier in the junctions container. This is a guideline rather than a hard rule.

C2.7 Notes

1) It is not expected that the MIS will store the Curriculum Structure in terms of “junctions” or use that term with its users. Instead, it is left to the MIS to re-interpret all these Junctions to **recreate** the Curricular Structure, and represent it in a way that is familiar to its users.

2) **Mixed years**: (eg in sixth-form) can be dealt with by defining the individual years first, then combining them to form the mixed-year group.

3) **Junction with only one output**: If a junction has only **one output** group then the default should be to assume that all the “input” students are in that output group.

4) There may be more than one way to describe a given curriculum using junctions. For eg, if a half-year-band is made-up from two Tutor-groups, then this could be described junction-wise as:

   Either: Divide year into half-year-bands, then:
   Input: half-year-band Output: two Tutor-groups

   Or; Divide year into its Tutor-groups, then:
   Input: two Tutor-groups Output: half-year-band

5) Not all MIS store the Curricular Structure, and so not all will be able to use the concept of the “Junction”. However, such MIS can simple ignore the Junctions and use the Teaching Groups (only) instead. This will still allow them to import the schedule – the only disadvantage is they will miss-out on some student eligibility rules.

6) The MIS should interpret the student eligibility rules in the Junctions in a “flexible / soft” way. That is, the rules are only a guide ... any individual student should be allowed to break them, if Office / Admin staff need to.

C**2.8 Further examples of Junction Models**

The Currciulum Structure varies greatly in complexity from school to school.

See **Appendix D** for more examples.

Object Definition C3:   Non-Teaching Activity

C3.1 Overview

A “non-teaching activity” is used to represent times when a teacher is not teaching a Teaching Group but is nonetheless “busy”.   Examples are meetings, part-time teachers who are not in school, PPA time, etc.

C3.2 Data Object Definition

Non-Teaching Activity			As described above (PPA, etc)	
	Element	**Char**	**Description**	**Type**
@	RefId	M	A unique GUID for this non-teaching activity	RefIdType
	NTCode	M	A short code for the non-teaching activity	xs:normalizedString maxLength = 12
	Description	M	A description of the non-teaching activity	xs:normalizedString maxLength = 25

	Type	M	Lesson/meeting type
			NB: This Type could differ from the Period Type in the **cycle** definition (for example, if Period 1 is Registration for some classes but not others)
				Value:
				R = registration
				B = break
				L = lunch
				O = outside school hours (eg: clubs)
				U = teacher unavailable (but in school)
				X = teacher unavailable (out of school)
				NB: **U** & **X** useful for cover-module or for further scheduling
				N = other (or unspecified) non-teaching activity

C3.3 Example XML:

```
<NonTeachingActivityList>
  <NonTeachingActivity RefId="000000000000315374616666726F6F6D">
    <NTCode>Staffroom</NTCode>
    <Description>Staffroom</Description>
    <Type>N</Type>
  </NonTeachingActivity>

  <NonTeachingActivity RefId="120000000000315374616666726F6443">
    <NTCode>PPA</NTCode>
    <Description>PPA time</Description>
    <Type>N</Type>
  </NonTeachingActivity>

  <NonTeachingActivity RefId="123400000000315374616666726F6901">
    <NTCode>UnAv</NTCode>
    <Description>Unavailable</Description>
    <Type>X</Type>
  </NonTeachingActivity>
Etc ... for all NTA’s
</NonTeachingActivityList>
```

Object Definition C4: Lesson

C4.1 Overview

The Schedule is described by listing all the "lessons" on the timetable. It contains the time that each lesson is taught, and the details of that lesson.

Note: "Lesson" here has a wider sense than simply a teacher teaching a TG, and may include meetings, PPA, after-school clubs, etc. In these cases, the "Type" will typically be 'N' (meaning "Non-teaching activity") and 'NTCode' will be provided rather than 'TeachingGroupID'.

C4.2 Data Object Definition

Lesson	A lesson scheduled on the timetable				
	Element	**Char**	**Description**	**Type**	Note
	TTDayId	M	Day this lesson is scheduled (see TTDayID)	IdRefType	
	PeriodIndex	M	Index of period this lesson/meeting starts	Integer	
	Type	M	Lesson/meeting type The value here determines whether a **TeachingGroupID** or an **NTCode** is required	Value: T = teaching N = non-teaching	
	Length	O	Length of lesson in periods (2=double-period lesson, etc). Only need to show for first lesson of the double. Omit if just a single.	Integer	[1]
	TeachingGroupRefId	C	The Teaching Group being taught (if Type = "T") (see TeachingGroup)	IdRefType	[2]
	NTCodeRef	C	If there is no TeachingGroupRefId (if Type = "N") (see NTCode)	IdRefType	[2]
	TeacherList	O		List	
	TeacherList/ TTTeacherRefId	MR	The teacher(s) teaching / supervising this lesson (see TTTeacher)	IdRefType	
	TTRoomRefId	O	The room (or location) where this lesson is taking place (see TTRoom)	IdRefType	

C4.3 Notes on Elements:

[1] MIS tend to prefer that we export a double as 2 lessons, but optionally you can still explain that the 2 singles are actually a double by assigning a <Length>2</Length> to the first lesson (meaning I am actually a double). Similarly for a treble; you export 3 lessons, but the 1st one would say <Length>3</Length>. Omit <Length> altogether if just a single.

[2] Each lesson must have either a **TeachingGroupRefId** (if it is a 'standard' lesson, with a TeachingGroup being taught, ie: Type = 'T') or an **NTCodeRefId** (if it is a non-teaching activity, ie: Type = 'N').

C4.4 Example XML:

```
<Schedule>
   <LessonList>
```

(In version 2 of the objects, other general information about this schedule could be stored here)

This Lesson is a teacher teaching a group of students for the **first** period of a ‘double’ lesson:

```
  <Lesson>
    <TTDayId>DY01</TTDayId>
    <PeriodIndex>1</PeriodIndex>
    <Type>T</Type>
    <Length>2</Length>
    <TeachingGroupRefId>AEECD89932678725A317B265CE015413</TeachingGroupRefId>
    <TeacherList>
      <TTTeacherRefId>1234567890123456789012345678900AB</TTTeacherRefId>
    </TeacherList>
    <TTRoomRefId>12345678901234567890123456789075</TTRoomRefId>
  </Lesson>
```

This Lesson is a meeting of 3 members of the Senior Management Team:

```
  <Lesson>
    <TTDayId>DY01</TTDayId>
    <PeriodIndex>1</PeriodIndex>
    <Type>N</Type>
    <NTCodeRef>Mtg</NTCodeRef>
    <TeacherList>
      <TTTeacherRefId>1234567890123456789012345678900CD</TTTeacherRefId>
      <TTTeacherRefId>A2345678901234567890123456789012</TTTeacherRefId>
      <TTTeacherRefId>B2345678901234567890123456789034</TTTeacherRefId>
    </TeacherList>
  </Lesson>

</Schedule>
```

C4.5 Part-time Teachers

Can be denoted by using “Type = N” above, and then referencing a Non-teaching Activity whose type = ‘X’ (teacher unavailable). For example, a teacher who is away on Mondays (which have, say, 5 periods in the day) could be represented by 5 lessons such as this:

```
  <Lesson>
    <TTDayId>DY01</TTDayId>
    <PeriodIndex>1</PeriodIndex>
    <Type>N</Type>
    <NTCodeRef>UnAv</NTCodeRef>     (“UnAv” has type ‘X’ in <NonTeachingActivityList>)
    <Length>5</Length>              (assuming Monday has 5 periods – only need to show this for 1st of 5 lessons)
    <TeacherList>
      <TTTeacherRefId>12345678901234222890123456789 0CD</TTTeacherRefId>
    </TeacherList>
  </Lesson>
```

C4.6 Publish the Whole Timetable Schedule

Even if there have only been some minor changes since the last timetable was published, it was agreed that the TT-Software should ***publish the whole timetable again***. (After some discussion, it was decided that an attempt to publish just the changes (even if possible) would be unsafe as it was subject to misinterpretation and could lead to confusion).

General Form of a Timetable Message

NB: Best way to send this message is probably going to be via ***SIF Zone Services***.

1 General form of our timetabling SIF messages

Each SIF message should include a header of general information (to “set the scene”) followed by data describing the Curricular Structure (and/or the Schedule, and/or the Basic Data).

XML Example:

```
<SIF_Message Version="x.y” xmlns="http://www.sifinfo.org/uk/infrastructure/2.x">

  <SIF_Event>
    <SIF_Header>
      <SIF_MsgId>AB34DC093261545A31905937B265CE01</SIF_MsgId>
      <SIF_Timestamp>2008-09-18T13:39:12-08:00</SIF_Timestamp>
      <SIF_SourceId>TimeTabler_October_Resolutions</SIF_SourceId>
    </SIF_Header>

    <SIF_ObjectData>

ObjectName could be: “CurricularStructure” or “BasicData” (only).
      <SIF_EventObject ObjectName="CurricularStructure" Action="Add">
        <CurricularStructure RefId="AB34DC093261545A31905937B265CE01">

          <Scope>
       Scope of this timetable ...
          </Scope>

          <BasicData>
            <TTTeacherList>
       List of teachers ...
            </TTTeacherList>
            <TTSubjectList>
       List of subjects ...
            </TTSubjectList>
            <TTRoomList>
       List of rooms ...
            </TTRoomList>
       (Optionally) list of sites (and in next version, students ?), etc ...
          </BasicData>

          <TeachingGroupList>
       List of teaching groups …
          </TeachingGroupList>

          <JunctionList>
       List of junctions …
          </JunctionList>

        </CurricularStructure>
      </SIF_EventObject>

    </SIF_ObjectData>
  </SIF_Event>
</SIF_Message>
```

2 Alternative to the above

Our timetable contains several different objects. If we do not want to send several different objects in the same message, then an alternative is to send several consecutive SIF messages which in total describe the timetable.

For example:

Message No.	Message contains
1	<Scope> with status = <BT>
2	<Basic Data>
3	<TeachingGroupList>
4	<JunctionList>
5	<NonTeachingActivityList>
6	<Schedule> (lessons)
7	<Scope> with status = <ET>

3 "Timetable data" Information to be included

Each case for sending a SIF message will include different amounts of information in that message, as follows:

Case	Information included in message
Basic Data (only)	Basic Data (teachers, subjects, rooms, etc)
Cycle	Basic Data (cycle)
Curricular Structure (only)	Basic Data, TeachingGroupList, JunctionList
Schedule	Basic Data, TeachingGroupList, JunctionList, NonTeachingActivityList, Schedule

4 Federation of Schools

Extra care needs to be taken when dealing with a **consortium / federation of schools**.

The possibilities for constructing the timetable for each school are:

a) Each school has its own timetabler, creating his own individual school timetable
This is currently the most likely option.

b) One timetabler does the timetable for all (or several) of the schools in the consortium
This may happen in some federations.

c) One timetable is devised to cover all the schools.
This is currently very unlikely, for many reasons (too many differences between the individual schools, insufficient resources to have everyone doing science at same time, constant & so disruptive movement between sites, etc). However, will become more possible with increase of virtual learning, etc.

However it is done, the timetablers at the various schools will need to consider agreeing to common times for their common activities (eg: the schools could reserve agreed mornings and afternoons, and move students and teachers only at lunch-time). Common option blocks would also need to be at the same time at each school, etc.

In each case, the following rules should be followed:

1. Even though they are in a federation, each individual school in the federation will still submit its timetable separately. (It is very unlikely that a single timetable could be devised to cover all schools, but even if it was, it would be submitted separately to each school).
2. Each individual school will need to include in its timetable a special "room" (location or "learning space") to represent each of the *other* schools in the consortium, for when students visit that other school. Teachers and students can then be assigned to that "room" to mean that they are at the other school. ie: you associate a

"room" with another school or site; one room per other school in the consortium. You may also wish to set-up separate "sites" for these rooms.

3. You may also need to have links to other schools or sites (eg: local colleges or businesses, or industrial locations) for when your students visit them. These links shall again be represented by "rooms" and "sites". *For more details; see the information on "Rooms" in the "Basic Data" section of this document.*

5 Obtaining the "SchoolInfoRefId" for a federation

To create the Header, the TT-Software will need to request the **"SchoolInfoRefId"** from the MIS.
But how does this work if there is a federation of several schools ?

If there is a consortium / federation of schools:
- Could handle this by the MIS & TT-Software having their own ***dedicated*** zone ?

If the same timetabler is creating the timetable for more than one school:
- Could target the correct zone for the school s/he's working on ?

This will be determined when we come to some real examples of federations.

Pilot Scheme

Piloting of the objects described in this document has been discussed. Vendors of both TT-Software and MIS are happy to engage in this pilot scheme, and initial discussions have already started.

This could be done either "off-line" (with test-data) and/or "on-line" at a real school.

It is expected that such a scheme will be set-up in early 2009.

Update: May 2009: Initial trials now in progress.

Appendices

Appendix A

List of Characteristic Codes

The Char(acteristics) column for all of the tables in this document use the standard SIF codes

Code	Characteristic
M	Mandatory element or attribute
O	Optional element or attribute
C	Conditional element or attribute
MR	Mandatory and repeatable element
OR	Optional and repeatable element
CR	Conditional and repeatable element

Elements marked “@” are attributes.

Appendix B

Names used for Groupings of Students

This is an additional note to provide more details about some of the key terms in the Glossary shown at:
http://www.timetabler.com/Glossary.pdf

There are at least 17 names used by timetablers for Groups of students in schools & colleges:
Registration Group, Form, Tutor Group, Curriculum-population, Basic Group, Class, Pure Class, Linear Group, Composite-class, Band, Half-year-group, Year-group, Teaching Group, Pupil Group (Student Group), Set, Set Group, Teaching Set.

They mainly group into 2 entities, of which examples are:

1. **7A**
2. **7A-French** (or 7A-French-Set1)

The notes on the following pages are intended to clarify their meanings.

Type 1 **7A is an example**

Registration Group is one of the least-ambiguous terms, and all schools have them, as the registration of students is a legal requirement. eg. 7A

Form has the same meaning and is also unambiguous, but now tends to be less used in maintained schools.

Tutor Group usually has the same meaning, but is strictly only relevant where there are pastoral care ‘tutorials’ for PSHE etc.

Curriculum-population is more long-winded but perhaps more accurate. It’s a group of students all having access to the same curriculum.

Basic Group has the same meaning, implying the basic grouping of students from which a Curriculum Diagram is constructed.

Class is also commonly used in this sense and is the term used in ***TimeTabler***. But ‘Class’ may be the most ambiguous of these words, with users sometimes meaning a RegistrationGroup (‘Class 7A’); sometimes a TeachingGroup (‘the 7A-French class’).

Pure class is a term used by timetablers to describe a (simple) curriculum where the lessons are taught to the whole ‘class’ (ie. the whole Registration Group) for different subjects.

Whichever of these terms is used, it is the basic unit needed for drawing the simplest rows on the Curriculum Diagram. This is the necessary starting point for each timetable (in order to describe the underlying curricular structure of the schedule).

Linear Group is a term used only in Nova (for a collection of ‘pure class’ activities).

Composite-class may be used (in schools where ‘class’ is used for a RegistrationGroup) in order to describe when 2 or more ‘classes’ come together. eg. classes 7A and 7B join together (as 7AB) for Technology.

Band is used when 2 or more RegistrationGroups are combined and scheduled together. eg. 7A, 7B, 7C join together as Band 7ABC for Games.

Half-year-group is a Band formed from (about) half of the Registration Groups in a Year.

Year-group (YearGroup) eg. Year 7 : all the students of that school age : all the RegistrationGroups which begin with ‘7’.

Type 2 **7A-French is an example**

Teaching Group is a group of students brought together for studying a particular Subject
[and is scheduled with a Teacher(s) in a Room(s) in a particular time-slot(s)].
eg. 7A-French

In some MIS it may be called a 'Class', in others it is called a 'Teaching Set'. But both these terms are problematic - 'Class' is ambiguous as it can be used in school to mean more than one thing (see above), while anything with the word 'Set' in it will imply 'setting' (see below).

TeachingGroup was the name agreed by the 2003 DfES Working Group.

A note about **Eligibility Rules** for populating TeachingGroups:

Pupil Group In the 2003 DfES Working Group this term was used for a group of students who may go to just one Teaching Group together, or may go to several different Teaching Groups together. Each PupilGroup 'serves' one or more Teaching Groups.

For example, if a Registration Group 7A is taught History together, but at other times of the week exactly the same students are also taught Geography together, and at other times they are taught Drama together, that would be one Pupil Group (7A), but 3 related Teaching Groups (7A/Hi1, 7A/Gg1, 7A/Dr1).

However, for other activities there may be one Pupil Group and one Teaching Group. For example, if the only time the pupils in 'Year 10 Maths set 1' get together is for 'Year 10 Maths set 1', then there would be one Pupil Group (called 10z/Ma1, say, ie. in Band z), and just one related Teaching Group (also called 10z/Ma1).

So the PupilGroup introduces the idea of 'student eligibility' and provides some basic 'Eligibility Rules'. It saves time for the office staff when they come to assign students to Groups in the MIS. Once the office staff have assigned the correct students to PupilGroup 7A, that automatically assigns the correct students to all 3 related Teaching Groups (7A/Hi1, 7A/Gg1, 7A/Dr1).

This suffices for many MIS, but some MIS require more advanced 'Eligibility Rules' (for example, to make it clear that a student in 10z/Ma1 cannot also be in 10z/Ma2, or that a student in 7A/Fr1 cannot also be in 7B/Fr1) and for that, a Curriculum Diagram is needed. A representation of the structure of the Curriculum Diagram needs to be exported to the MIS, so the MIS understands how 7A, 7B, 7x, etc relate to one another.

There is more information about Curriculum Diagrams at :
http://www.timetabler.com/SupportCentre/CurriculumDiagram.xls

Set,
Set Group,
Teaching Set
eg. 7A-French-Set1

These are all special cases of a **Teaching Group**.
Any term including 'Set' should only be used for a Group that has been 'setted' ie. has been determined by ability in that Subject (or more correctly, determined by prior attainment in that Subject).
eg. 7ABC-Maths-Set1, 7ABC-Maths-Set2, 7ABC-Maths-Set3, etc.

In addition there are groupings for scheduling into the same time-slot(s) called (interchangeably) **Block, Pool, Column, Options-Block, Options-Pool, Options-Column** [with the word **'Electives'** also being used in place of 'Options'].

A **Container Block** can be used to constrain a number of TeachingGroups into common time-slots.

And **Circus, Rota, Rotation, Carousel, Cyclic Block** are used (interchangeably) to describe how the staff may organise the composition of the TeachingGroups during the calendar year.

For the SIFA group, on this topic, we think the 2 key points are:

1. We need a term to describe the simplest grouping of students, for constructing a Curriculum Diagram. eg. 7A
   There are 6 choices above (Registration Group, Form, Tutor Group, Curriculum-population, Basic Group, Class).
2. We need a term to describe a group of students brought together to be taught a particular Subject. eg. 7A-French
   TeachingGroup is the obvious and natural choice for this.

Appendix C

US & Australian SIF Timetabling Models

The US and Australian models were considered by the Task Force, along with the DfES XML Specification for the transfer of timetabling information, from back in 2003.

US SIF Model:

The US model seems totally different from how timetables are scheduled in the UK.

Australian SIF Model:

The Australian model seems to be closer to the UK method, but with a different emphasis. We are in contact with the Australian SIF Timetabling team, and currently seeing what common ground there is.

DfES Spec 2003:

The (then) DfES specified an XML model for the transfer of timetabling information back in 2003. This included the concept of "Pupil Groups" as well as "Teaching Groups", and has been used since that time to transfer timetables between TimeTabler and Phoenix e1, using XML.

Appendix D

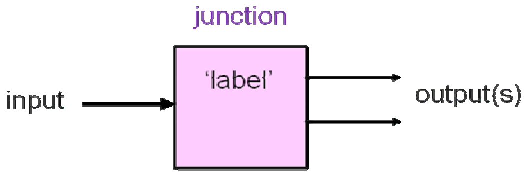
Curriculum Diagrams and the Junction Model

The Curriculum Structure is represented by "Junctions".   For examples of the Junction Model, see the slides presented at the Curriculum & Timetabling Task Force Meeting in Bangor on 8 September 2008.
They were used as the basis of a discussion on how to transfer information about curricular structures and the 'eligibility rules' which are inherent in them.

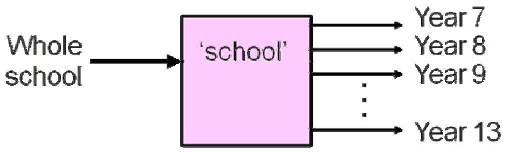
They show a range of Curriculum Diagrams, from the simple to the complex, in use in schools today.
Paired with each Curriculum Diagram is a 'Junction Diagram' to show how the curricular structure and eligibility could be exported, for example from a timetabling program to an MIS.

The conclusion was that the Junction Model deals well with the whole range of curricular structures.

The Style of the Junction Diagrams is like this:



Simple Example splitting the school into its years:



Two examples of Curricula and their Junction Models are given on the next two pages.

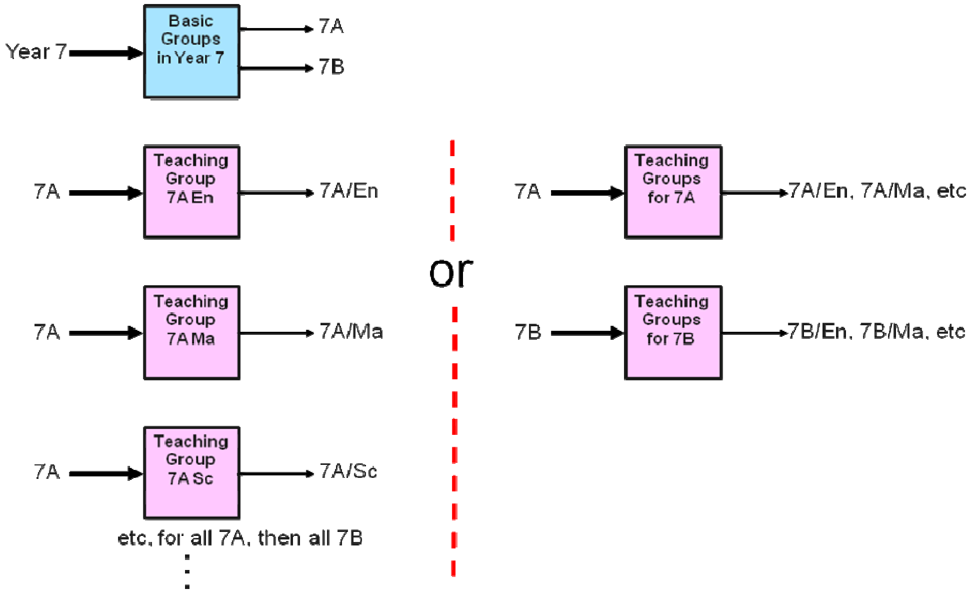
For further examples, and a range of Curricula, see the document "***SIFA-CurriculumDiagrams-JunctionDiagrams.pdf***" on the Curriculum & Timetabling Discussion Forum.

Example A (Simple example):
Curriculum Diagram:

2 form entry (2 f.e.) All 'pure class' activities :

7A	En	Ma	Sc	Fr	Hi	Gg	Ar	Mu	RE	SE	Te	PE
7B	En	Ma	Sc	Fr	Hi	Gg	Ar	Mu	RE	SE	Te	PE

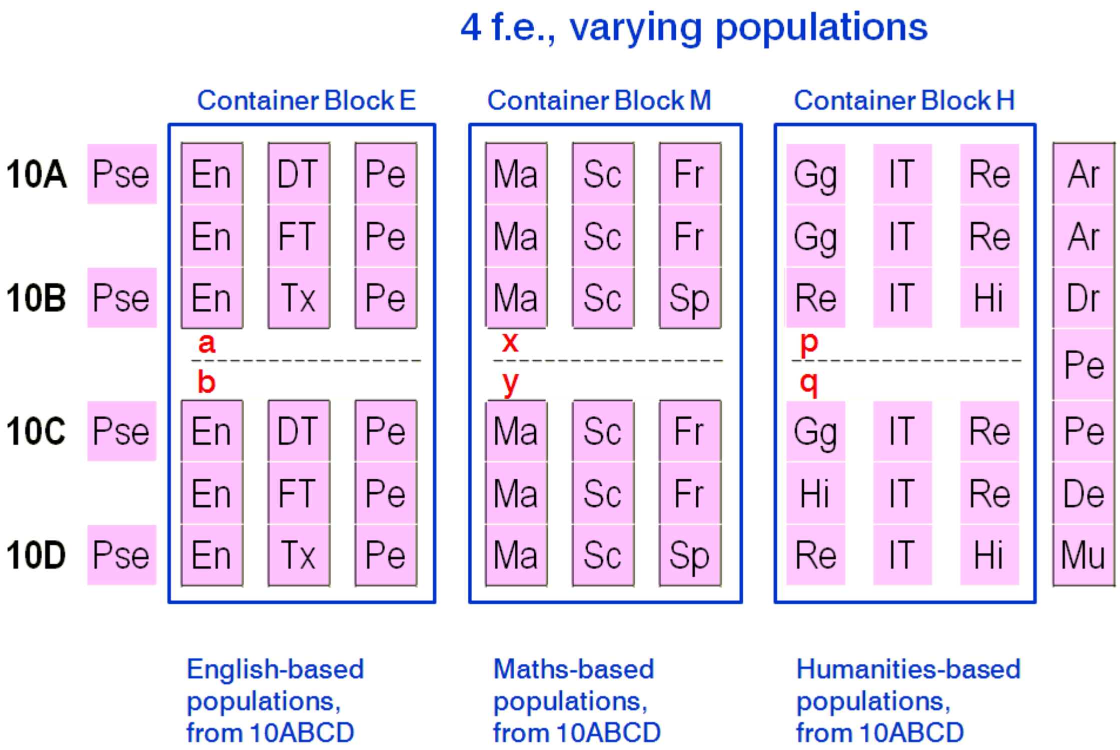
Can be represented by this Junction-Model:



Note: The above shows two possible ways of exporting 7A's TGs, as discussed at Bangor meeting.

Consensus was that we would use LHS method rather than RHS, and the objects are defined on that basis.

Example B (Complicated example):
Curriculum Diagram:



4 f.e., varying populations

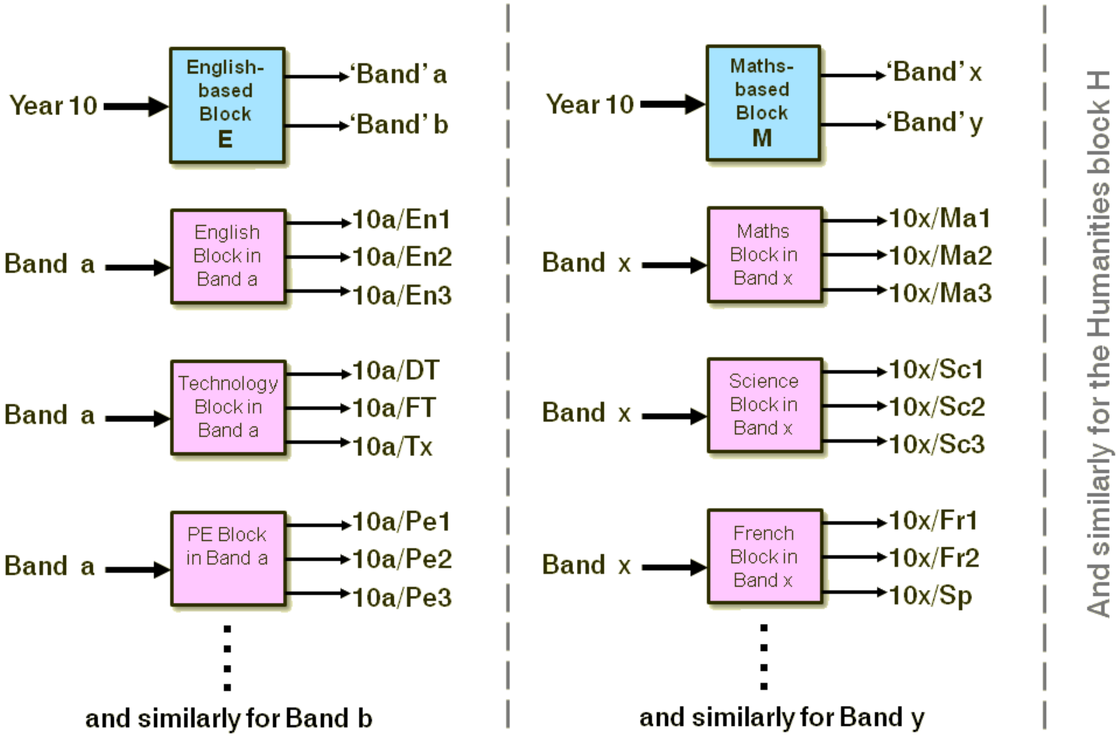
		Container Block E			Container Block M			Container Block H			
10A	Pse	En	DT	Pe	Ma	Sc	Fr	Gg	IT	Re	Ar
		En	FT	Pe	Ma	Sc	Fr	Gg	IT	Re	Ar
10B	Pse	En	Tx	Pe	Ma	Sc	Sp	Re	IT	Hi	Dr
		a			x			p			Pe
		b			y			q			
10C	Pse	En	DT	Pe	Ma	Sc	Fr	Gg	IT	Re	Pe
		En	FT	Pe	Ma	Sc	Fr	Hi	IT	Re	De
10D	Pse	En	Tx	Pe	Ma	Sc	Sp	Re	IT	Hi	Mu

English-based populations, from 10ABCD

Maths-based populations, from 10ABCD

Humanities-based populations, from 10ABCD

Can be represented by this Junction-Model:



Year 10 → English-based Block E → 'Band' a, 'Band' b

Band a → English Block in Band a → 10a/En1, 10a/En2, 10a/En3

Band a → Technology Block in Band a → 10a/DT, 10a/FT, 10a/Tx

Band a → PE Block in Band a → 10a/Pe1, 10a/Pe2, 10a/Pe3

⋮

and similarly for Band b

Year 10 → Maths-based Block M → 'Band' x, 'Band' y

Band x → Maths Block in Band x → 10x/Ma1, 10x/Ma2, 10x/Ma3

Band x → Science Block in Band x → 10x/Sc1, 10x/Sc2, 10x/Sc3

Band x → French Block in Band x → 10x/Fr1, 10x/Fr2, 10x/Sp

⋮

and similarly for Band y

And similarly for the Humanities block H

Appendix E

Members of the Task Force

At the time of this document, the regular members of the Curriculum & Timetabling Task Force are:

- Andy Ferguson (*RM*)
- Chris Johnson (*TimeTabler*)
- David Stott (*Capita*)
- John Ashworth (*Pearson-Phoenix*)
- John Story (*Semantise*) (Chair)
- Keith Johnson (*TimeTabler*)
- Oggy East (*Semantise*)

Other interested parties (eg; *Serco*) have kept in touch with progress via email and/or telecons.

The ongoing work of the Task Force can be viewed on the Curriculum & Timetabling SIFA Community Site at:

http://community.sifinfo.org/sites/InternationalConsortiums/UnitedKingdom/DataStandards/Lists/Curriculum%20%20Timetabling%20discussion%20forum/AllItems.aspx

The group welcomes all feedback; please post your suggestions or questions to this site.

Appendix F

SIF Services

"SIF Services" may be a better way forward than "SIF Objects".
We intend to investigate this once "SIF Services" are ready / mature.

Appendix G

Modifications since v1.0

Preparation for testing of the SIF objects commenced during April / May 2009, by looking at sending a timetable from the *TimeTabler* timetabling program to the *RM-Integris-G2* MIS. During this testing, the following changes to v1.0 of the objects were found necessary or desirable for a successful transfer:

Changes from v1.0 to v1.1	Reason
Junction: OutputList/Output/**GroupLocalId** changed from 'M'andatory to 'C'onditional	Not required if Type = "TG"
It's now made clearer that a **Junction: Type** of "YR", although typically a year, can in fact represent any "*principal division*" of the school	The "*principal division*" of the school could be a combination of years, or in theory, any other population of students.
Junction: OutputList/Output/**NCYear** now becomes **NCYearList**	The junction may represent a population which is a combination of two or more years. NB: the element remains optional.
Junction: **Type** renamed to **LocalType**	Keeps the following meanings consistent: **Type** = choose from a defined list of values **LocalType** = a free-format type for which different suppliers can choose their own values.
Junction: InputList/Input/**Type** is now conditional and has only one allowed value: "SC"	Type only required for "SC" case; "SC" has special meaning – it means this is the top of the "tree" and the input is the whole school.
There is now a **Junction: ShortLabel** as well as a **Junction: Label**	For display purposes.
Scope: <SchoolYear> is now the year the school year *starts* in (not ends).	More natural way to think of it.
Non-teaching activities now have a **RefId** (not just a code)	To be consistent with other similar objects.
TimetableTeacher object renamed as **TTTeacher**.	Makes XML code more legible (and shorter !); repetition of "Timetable" throughout BasicData was distracting & unnecessary.
Ditto for other BasicData objects: **TimetableSubject → TTSubject** **TimetableRoom → TTRoom**	As above.
Scope: SchoolInfoRefId no longer marked as @	It shouldn't be an 'attribute'.
Ditto for these **TeachingGroup** elements: **MainTeacherRefId** GroupTeacher**TTTeacherRefId**	As above.
Room: SiteId renamed **SiteRefId**	For consistency with other IdRefType types.
Period: StartTime and **EndTime** rules described more explicitly	For sake of clarity.
TeachingGroup: RefId is now a GUID	For consistency with other objects.

Junction: RefId removed	There is no need for it … we should always send the complete “tree” of junctions. Trying to change one junction in the middle of the “tree” in isolation could be dangerous.
Junction: ShortLabel is now ‘M’andatory	Useful for screen displays.
xs:normalizedString elements: ‘max lengths’ now suggested	Helps TT-Software and MIS to know what to expect, and reduces need for truncation and/or warnings.
OutputList/Output/ NCYearList/**NCYear** has **‘char’** changed to ‘MR’ (mandatory repeat). Ditto for similar elements.	To comply with SIF notation (repeated elements within a list that must appear if the list appears).
OutputList /Output/**Desc** changed from ‘M’andatory to ‘C’onditional	Not required if Type = “TG”, because a Teaching Group has its own <ShortName> and <LongName>
TTDayList/TTDay/**PeriodList** changed from ‘M’andatory to ‘O’ptional	<PeriodList> not required if this day has the same periods as the ‘standard day’.
Lesson: **TTSubjectRefId** removed	The TeachingGroup already has a subject, which we don’t want over-ruling for a specific lesson, as it can cause confusion. Also, we don’t want non-teaching activities to have a subject.
The meaning of Lesson: **Length** has been modified	MIS tend to prefer that we export a double as two separate (single) lessons
(MIS)Room removed from list of objects – may be defined later if needed (see section B6)	This document should only concern itself with objects specifically related to timetabling

Changes from v1.1 to v1.2	**Reason**
Junction: OutputList /Output/**ShortDesc** added	For display purposes
Scope: ScheduleName is now ‘M’andatory	For display purposes & to help differentiate schedules
Period: ShortName is now ‘C’onditional	Only *Mandatory* when **Period** appears in StandardPeriodList. Not required when used in TTDayList/TTDay/PeriodList
Period: LongName is now ‘C’onditional	Same reasoning as above
“Junction Rules & Relationships” have been spelt-out in the section on Junctions	To clarify how junctions work

Changes from v1.2 to v1.3	**Reason**
XML examples have now been aligned with the latest Object Definitions	Some changes to definitions had not been reflected in the examples

NB: Any further proposed changes to this Spec should fit into these SIF timescales:

Timescales (2009):

27th March — final date for submissions
1st May — Level 1 Testing to have been passed on all proposed objects
14th May — 30-day public consultation period begins
12th June — 30-day public consultation period concludes
29th June — SIF Implementation Specification