

International Telecommunication Union

ITU-T

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

Z.601

(02/2007)

SERIES Z: LANGUAGES AND GENERAL SOFTWARE
ASPECTS FOR TELECOMMUNICATION SYSTEMS

Middleware – Distributed processing environment

Data architecture of one software system

ITU-T Recommendation Z.601



International
Telecommunication
Union

Documents
Document Sharing Hub

ITU-T Z-SERIES RECOMMENDATIONS
LANGUAGES AND GENERAL SOFTWARE ASPECTS FOR TELECOMMUNICATION SYSTEMS

FORMAL DESCRIPTION TECHNIQUES (FDT)	
Specification and Description Language (SDL)	Z.100–Z.109
Application of formal description techniques	Z.110–Z.119
Message Sequence Chart (MSC)	Z.120–Z.129
Extended Object Definition Language (eODL)	Z.130–Z.139
Testing and Test Control Notation (TTCN)	Z.140–Z.149
User Requirements Notation (URN)	Z.150–Z.159
PROGRAMMING LANGUAGES	
CHILL: The ITU-T high level language	Z.200–Z.209
MAN-MACHINE LANGUAGE	
General principles	Z.300–Z.309
Basic syntax and dialogue procedures	Z.310–Z.319
Extended MML for visual display terminals	Z.320–Z.329
Specification of the man-machine interface	Z.330–Z.349
Data-oriented human-machine interfaces	Z.350–Z.359
Human-machine interfaces for the management of telecommunications networks	Z.360–Z.379
QUALITY	
Quality of telecommunication software	Z.400–Z.409
Quality aspects of protocol-related Recommendations	Z.450–Z.459
METHODS	
Methods for validation and testing	Z.500–Z.519
MIDDLEWARE	
Distributed processing environment	Z.600–Z.609

For further details, please refer to the list of ITU-T Recommendations.

ITU-T Recommendation Z.601

Data architecture of one software system

Summary

ITU-T Recommendation Z.601 identifies a set of data structures and formats of one software system. These data forms appear at the various interfaces to and media of the system and comprise intermediate forms for transformations between the external forms. The data forms are needed within one system, and they are not abstracted away from the system over several systems.

This Recommendation identifies data schemata that may be used to define interfaces between software components. However, this Recommendation defines no software architecture.

Source

ITU-T Recommendation Z.601 was approved on 13 February 2007 by ITU-T Study Group 17 (2005-2008) under the ITU-T Recommendation A.8 procedure.

Keywords

Architecture, data, form, format, framework, layer, population, process, schema, structure.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

Compliance with this Recommendation is voluntary. However, the Recommendation may contain certain mandatory provisions (to ensure e.g. interoperability or applicability) and compliance with the Recommendation is achieved when all of these mandatory provisions are met. The words "shall" or some other obligatory language such as "must" and the negative equivalents are used to express requirements. The use of such words does not suggest that compliance with the Recommendation is required of any party.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementers are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database at <http://www.itu.int/ITU-T/ipr/>.

© ITU 2007

All rights reserved. No part of this publication may be reproduced, by any means whatsoever, without the prior written permission of ITU.

CONTENTS

Page

1	Scope	1
2	References.....	1
3	Definitions	1
4	Abbreviations.....	3
5	Conventions	4
6	Data architecture	4
6.1	Three-schema architecture.....	4
6.2	Seven-schema architecture	6
6.3	Communication and distribution	9
6.4	Nesting.....	11
Annex A – Requirements on schema notations		13
A.1	Introduction	13
A.2	Requirements on notations for the external terminology schemata	13
A.3	Requirements on notations for the concept schemata	17
A.4	Requirements on notations for the contents schemata	18
A.5	Requirements on notations for the layout schemata.....	19
A.6	Requirements on notations for internal terminology schemata	20
A.7	Requirements on notations for the distribution schemata	20
A.8	Requirements on notations for the physical schemata	21
A.9	Requirements on notations for the system management schemata	21
Appendix I – Introduction to data architectures.....		22
I.1	A system planning perspective	22
I.2	A data perspective on a system	24
I.3	Communication between systems	26
I.4	Communicating processes	27
I.5	Separation of media	27
Appendix II – Comparison with other architectures		29
II.1	Comparison with ITU-T Rec. M.3020	29
Bibliography.....		30

Introduction

The primary users of this Recommendation will be software developers who design data definitions of a system and its interfaces.

Data architects and systems planners may use this Recommendation to coordinate definitions of interfaces between systems.

Some of the data definitions may provide end users' understanding of the system and its functionality. Hence, these definitions will provide a kernel of a contract between purchaser and developer organizations.

Formal language designers may use this Recommendation to identify the scope of their notation and identify features needed to cover a certain application domain.

Annex A provides requirements on schema languages.

Appendix I provides an introduction and context of the Recommendation, where a system is defined in the context of systems planning of several systems. The reader is advised to read this appendix before he reads clause 6.

Appendix II provides a comparison with [b-ITU-T M.3020].

ITU-T Recommendation Z.601

Data architecture of one software system

1 Scope

This Recommendation identifies a set of data structures and formats of one software system.

These data forms appear at the various interfaces to and media of the system and comprise intermediate forms for transformations between the external forms.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

[ITU-T Z.351] ITU-T Recommendation Z.351 (1993), *Data oriented human-machine interface specification technique – Introduction*.

3 Definitions

This Recommendation contains a set of definitions. The following terms appear in this Recommendation. They are either defined explicitly, or by simple inference from the definitions. The terms are meant to be used within the context of the data architecture, and may not have the same meaning outside this context.

application area: See [ITU-T Z.351].

application layer: See [ITU-T Z.351].

application population: See [ITU-T Z.351].

application process: See [ITU-T Z.351].

application schema: See [ITU-T Z.351].

concept layer.

concept population.

concept process.

concept schema.

contents layer.

contents population.

contents process.

contents schema: See [ITU-T Z.351].

distribution layer.

distribution population.

distribution process.

distribution schema.

external layer: See [ITU-T Z.351].
external population: See [ITU-T Z.351].
external process: See [ITU-T Z.351].
external schema: See [ITU-T Z.351].
external terminology layer.
external terminology population.
external terminology process.
external terminology schema.
internal layer: See [ITU-T Z.351].
internal population.
internal process.
internal schema.
internal terminology layer.
internal terminology population.
internal terminology process.
internal terminology schema.
layer: See [ITU-T Z.351].
layout layer.
layout population.
layout process.
layout schema: See [ITU-T Z.351].
physical layer.
physical population.
physical process.
physical schema.
population: See [ITU-T Z.351].
process: See [ITU-T Z.351].
schema: See [ITU-T Z.351].
system.
system management layer.
system management population.
system management process.
system management schema.
terminology layer.
terminology population.
terminology process.
terminology schema.

4 Abbreviations

This Recommendation uses the following abbreviations:

AL	Application Layer
AP	Application Population
Ar	Application Process
AS	Application Schema
CL	Contents Layer
CP	Contents Population
Cr	Contents Process
CS	Contents Schema
DL	Distribution Layer
DP	Distribution Population
Dr	Distribution Process
DS	Distribution Schema
EL	External Layer
EP	External Population
Er	External Process
ES	External Schema
eTL	external Terminology Layer
eTP	external Terminology Population
eTr	external Terminology Process
eTS	external Terminology Schema
IL	Internal Layer
IP	Internal Population
Ir	Internal Process
IS	Internal Schema
iTL	internal Terminology Layer
iTP	internal Terminology Population
iTr	internal Terminology Process
iTS	internal Terminology Schema
LL	Layout Layer
LP	Layout Population
Lr	Layout Process
LS	Layout Schema
OL	Concept Layer
OP	Concept Population

Or	Concept Process
OS	Concept Schema
PL	Physical Layer
PP	Physical Population
Pr	Physical Process
PS	Physical Schema
SML	System Management Layer
SMP	System Management Population
SMr	System Management Process
SMS	System Management Schema
TL	Terminology Layer
TP	Terminology Population
Tr	Terminology Process
TS	Terminology Schema

5 Conventions

S Schema

P Population

6 Data architecture

6.1 Three-schema architecture

The three-schema architecture comprises the following kinds of schemata:

- external schemata;
- application schema;
- internal schemata.

For one system instance, the three-schema architecture provides one centralized application schema, which may support several external and internal schemata. For n external and m internal schemata, the three-schema architecture provides $(n+m)$ mappings between schemata.

The application schema of a system instance defines the data classes, constraints and derivations which have to be enforced for any data within the system instance, independently of which external or internal medium they may appear on.

Figure 1 depicts example schemata and corresponding populations in the three-schema architecture.

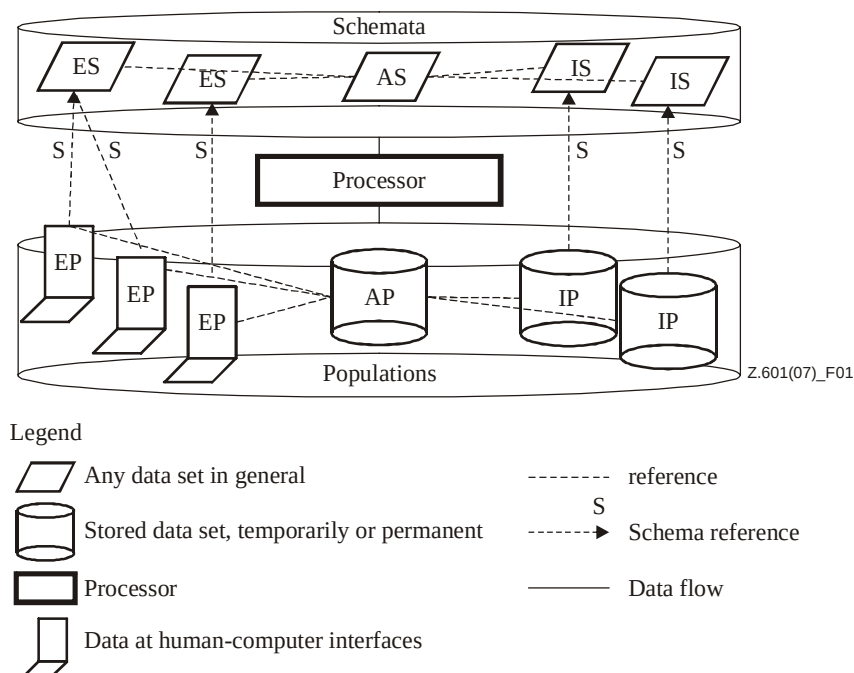


Figure 1 – Example use of the three-schema architecture

Note that the three-schema architecture does not state which external population and corresponding schemata are used to input data to which internal population and corresponding schema. Hence, internal forms are decoupled from external forms of data. Also, each external and internal schema is decoupled from any other external or internal schema, respectively. The application schema provides the only link between external and internal forms.

The correspondences between schemata, processors and populations may be conveyed through the notion of layers:

- external layer;
- application layer;
- internal layer.

This is illustrated in Figure 2.

Note that the definition of the application schema presupposes that every data instance is transformed via the application layer. Strictly speaking, the external form of data is one instance, the application form is a second and the internal form is a third. And these may be copied into several external and internal populations, creating still more instances, all originating from one input instance, e.g., in the external layer.

Note that the three-schema architecture is a compiler kind of architecture, where presentation forms of data at the external and internal layers are compiled via an intermediate application layer. This compiler feature of the architecture may be used to nest the architecture.

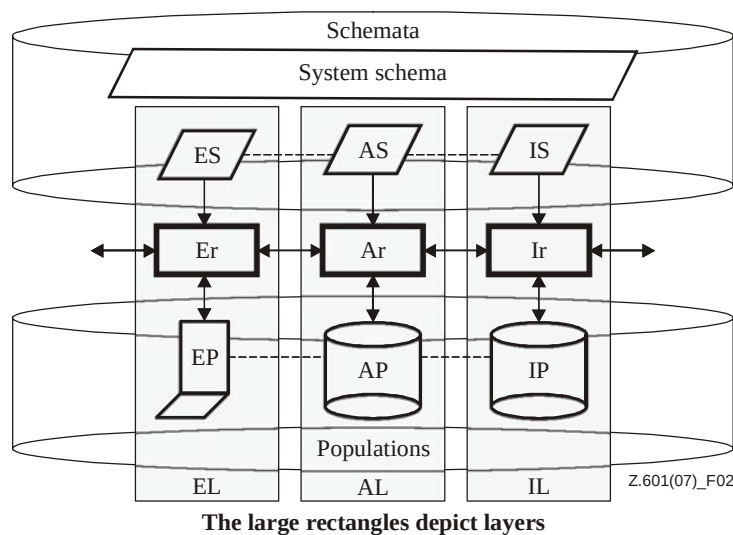


Figure 2 – Layering

In order to cover the coordination between the layers, we introduce some extra notions:

- System management schema, containing – except the external, application and internal schemata:
 - security data, including access control data;
 - system directory data, including data for configuration control.
- System management processor, including the external, application and internal processors, controls their interoperation and provides security and directory services.
- System management population: provides data instances for security and directory services.

Note that the system management processor and population are not depicted in Figure 2. With the system management schema, we have in fact a three plus one-schema architecture.

We also observe that the processor of the system instance is split into specialized processors per layer.

6.2 Seven-schema architecture

The seven-schema architecture provides a detail of the three-schema architecture. Each layer of the three-schema architecture is split into two or three sub-layers. This is depicted in Figure 3.