FDSN WG-2 Data formats and Data centers
IUGG 2015, Prague, Czech Republic
June 27, 18:00 – 20:00
PCC Floor 3, South Hall

StationXML

- status
- StationXML possible extensions
- representation negative normalization factor

SEED

- dataless SEED: extended support or end-of-life ?
- mini-SEED limitations/extensions: how to cope withe these ?

QC metrics

- definitions EIDA, IRIS (MUSTANG); FDSN standard ?

QuakeML

- status (Fabian Euchner)

- And all the other fields in station.txt not in StationXML

WG II

- StationXML extensions
  - network code extension
  - data availability
  - metadata versioning (date-time suggestion)
    - critical metadata only perhaps?
  - waveform versioning (data-time suggestion)
- Extensions to miniSeed
  - To be addressed:
    - Expanded network field
    - Byte order, record length, encoding, microseconds, timing quality in fixed record header
    - Forward compatibility mapping, documented procedure
  - To be explored and potentially addressed:
    - floating-point data compression encoding
    - fixed-point data encoding. With compression?
    - minimal format (microSEED?) or considerations for low latency real-time data flow?
      - shorter record lengths for low latency applications?
      - variable length records?
- QA Metrics XML
- Position on Infrasound
- Should we attempt to get an FDSN Data Center Cloud Presence
- Virtual network support at FDSN Data Centers
- Feedback on need for JSON metadata support, DMC has a strawman

StationXML status

From collaborative work by SCEDC/Caltech, GEOFON/GFZ, NCEDC/Berkeley, IRIS DMC to FDSN accepted standard in Nov. 2012

- GitHub repository (master branch)
- procedure for modifications: FDSN WGII
- documentation (basic)
- in operation at all large datacenters (e.g ORFEUS EIDA, IRIS DMC; all stations metadata available as StationXML

Important issues:

- full documentation (cf. SEED) in schema (SEED manual in html code available ?)
- StationXML builder, editor and management software (PDCC ?)

IRIS Data Management Center: http://service.iris.edu
fdsnws-station: http://service.iris.edu/fdsnws/station/1
Northern California Earthquake Data Center: http://service.ncedc.org
fdsnws-station: http://service.ncedc.org/fdsnws/station/1
BGR Hannover, Germany: http://eida.bgr.de/
fdsnws-station: http://eida.bgr.de/fdsnws/station/1/
Boğaziçi University, Kandilli Observatory: http://www.koeri.boun.edu.tr/2/tr/
fdsnws-station: http://eida-service.koeri.boun.edu.tr/fdsnws/station/1/
ETHZ: http://arclink.ethz.ch/fdsnws/
fdsnws-station: http://arclink.ethz.ch/fdsnws/station/1/
GEOFON Program, GFZ: http://geofon.gfz-potsdam.de/
fdsnws-station: http://geofon.gfz-potsdam.de/fdsnws/station/1/
IPGP Data Center: http://centrededonnees.ipgp.fr
fdsnws-station: http://eida.ipgp.fr/fdsnws/station/1/
INGV: http://webservices.rm.ingv.it/
fdsnws-station: http://webservices.rm.ingv.it/fdsnws/station/1/
LMU Munich, Germany: http://www.geophysik.uni-muenchen.de/observatory/seismology
fdsnws-station: http://erde.geophysik.uni-muenchen.de/fdsnws/station/1/
NIEP, Romania: http://www.orfeus-eu.org/eida/eida_niep.html
fdsnws-station: http://eida-sc3.infp.ro/fdsnws/station/1/
ORFEUS Data Center: http://www.orfeus-eu.org/
fdsnws-station: http://www.orfeus-eu.org/fdsnws/station/1/
RESIF: http://portal.resif.fr/
fdsnws-station: http://ws.resif.fr/fdsnws/station/1/
USP Seismological Center, Brazil: http://www.moho.iag.usp.br/
fdsnws-station: http://seisrequest.iag.usp.br/fdsnws/station/1/

Current (known) webservices providing StationXML

http://www.fdsn.org/xml/station/

FDSN StationXML Schema

The purpose of the FDSN StationXML schema is to define an XML representation of the most important and commonly used structures of SEED 2.4 metadata.

The goal is to allow mapping between SEED 2.4 dataless SEED volumes and this schema with as little transformation or loss of information as possible while at the same time simplifying station metadata representation when possible. Also, content and clarification has been added where lacking in the SEED standard.

When definitions and usage are underdefined the SEED manual should be referred to for clarification.

Another goal is to create a base schema that can be extended to represent similar data types.

Downloads

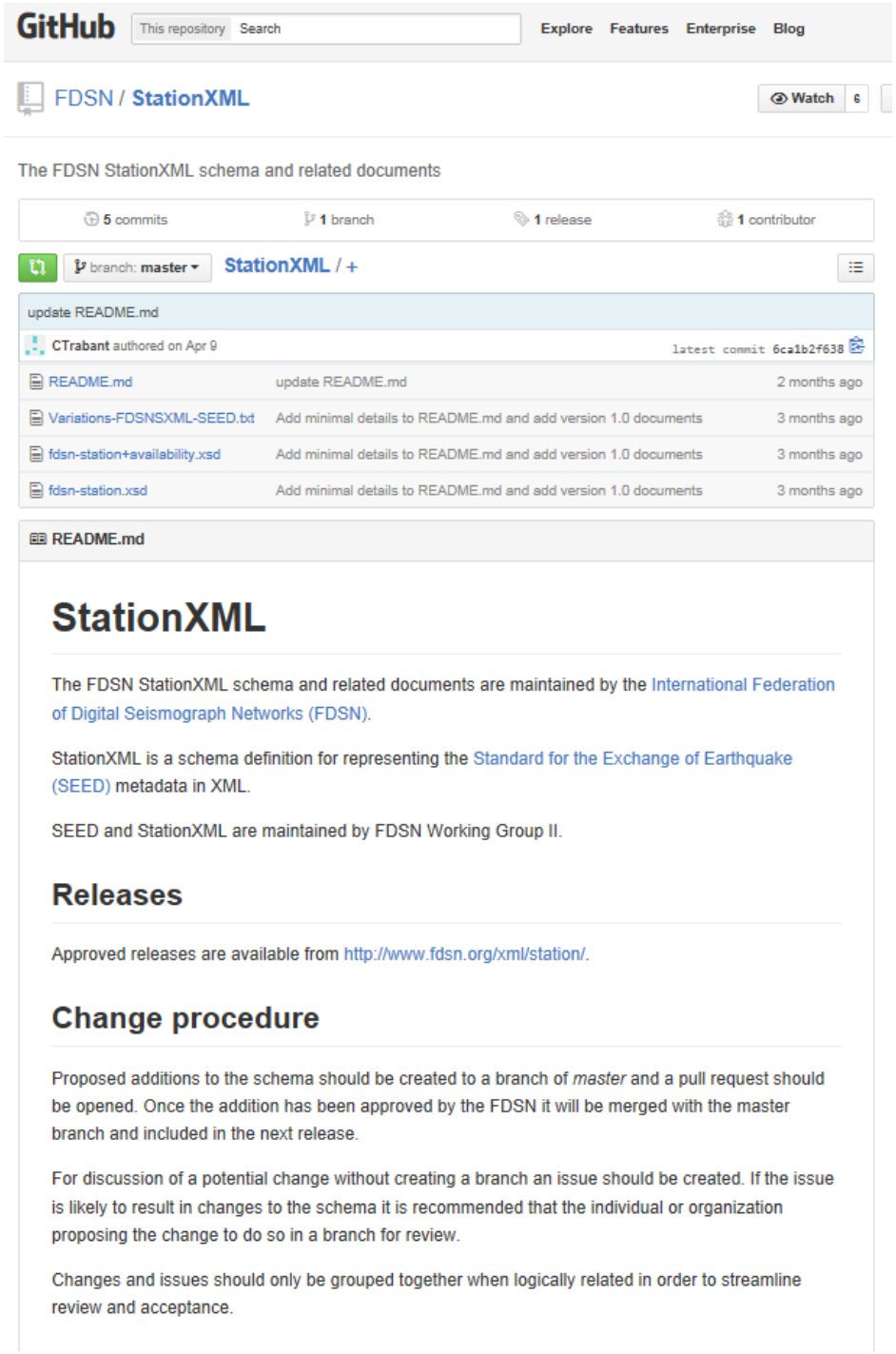
fdsn-station-1.0.xsd
Base FDSN StationXML schema

fdsn-station+availability-1.0.xsd
Extension of the base schema that includes time series data availability structures

Variations-FDSNSXML-SEED.txt
An overview of major variations between SEED 2.4 and FDSN StationXML

Examples will be provided in the future.

StationXML Versioning



Schema repository:

https://github.com/FDSN/StationXML

- FDSN StationXML schema extension by collaborative effort.

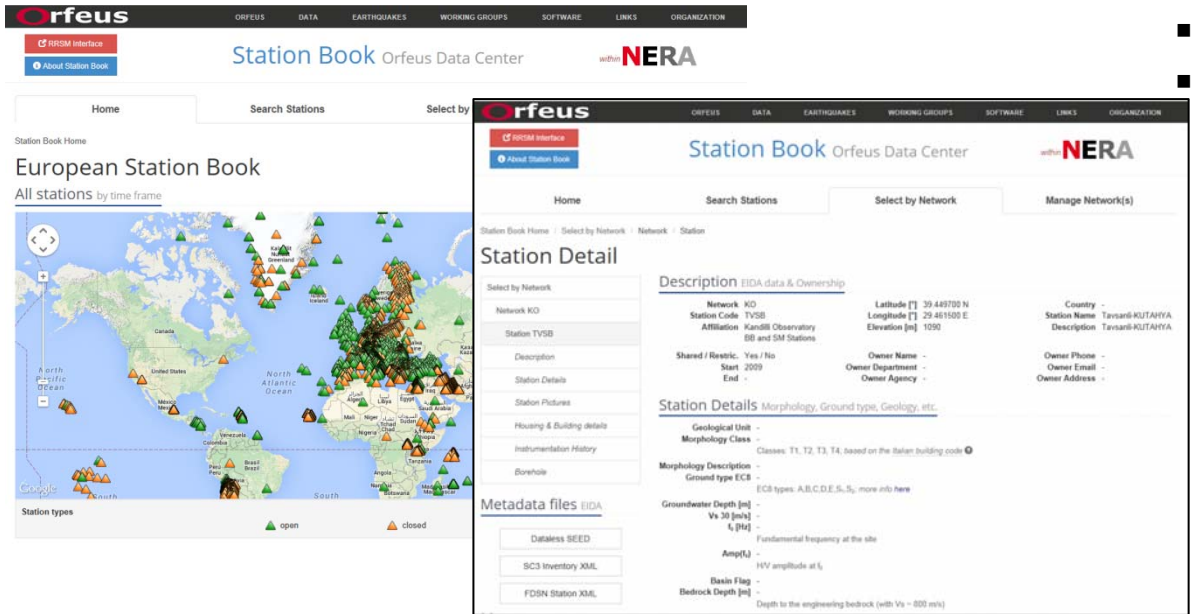
- gatekeeper for incorporating WGII approved changes

- create modifications of the schema as branches.

StationXML required/possible extensions

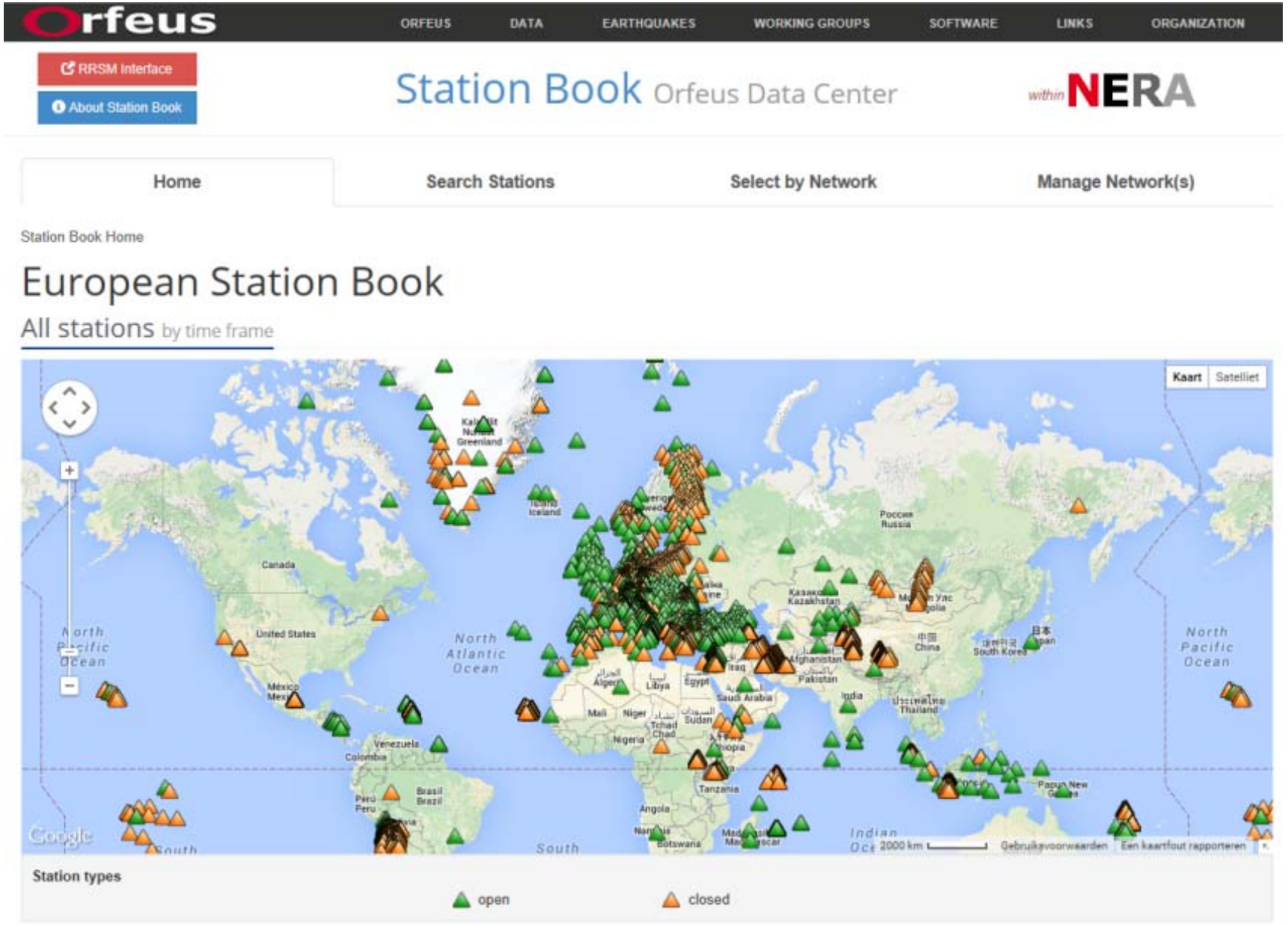
- network code extension (3 characters ?) - > affects mini-SEED and compatibility with dataless SEED
- data availability (Tim)
- metadata and waveform versioning (Tim; date-time suggestion ?)
- other (input required):
  - FDSN WGI StationBook
  - SM specific information (e.g. site characteristics, geology, soil conditions, etc.; ODC/ETH StationBook example (NERA); SC3 database extension; database ingested from EIDA and interactive by network operator; web interface ready – WGI ?)

www.orfeus-eu.org/stationbook



- geological units
- station pictures
- borehole information
- engineering parameters
  - morphology class (Italian building codes)
  - morphology description
  - ground type (EC8 types)
  - V_s 30, f_0
  - housing & building details (e.g. housing class)

www.orfeus-eu.org/stationbook



Station Book Orfeus Data Center

within NERA

Network Detail

Description EIDA data & Ownership

Network Code	KO	**Institutions**	-	**Archive**	KOERI
Region	-	**Type**	-	**Start**	1980
Description	Kandilli Observatory BB and SM Stations	**Class / Restric.**	P / No	**End**	-
Owner Name	-	**Owner Phone**	-		
Owner Department	-	**Owner Email**	-		
Owner Agency	-	**Owner Address**	-		

Metadata files EIDA

Dataless SEED

SC3 Inventory XML

FDSN Station XML

Stations EIDA data & Details (geology, morphology, EC8, Vs 30, housing, building, etc).

Station types

▲ open ▲ closed

Showing **1** to **10** of **183** rows

<< Previous | Next >>

Display 10 rows

Net	Sta	Lat [°]	Lon [°]	Elev [m]	Open	Rest.	Country	Station Name	Affiliation	Start	End	Info
KO	ADVT	40.4332 N	29.7383 E	193	Yes	No		Abdulvahap-Iznik-BURSA	Kandilli Observatory BB and SM Stations	2006	-	details
KO	AFSR	39.4468 N	33.0707 E	1055	Yes	No		Afsar-ANKARA	Kandilli Observatory BB and SM Stations	2008	-	details
KO	AGRB	39.5755 N	42.9920 E	1820	Yes	No		AGRI	Kandilli Observatory BB and SM Stations	2006	-	details
KO	AKDM	38.3285 N	42.9800 E	1662	Yes	No		Akdamar-VAN	Kandilli Observatory BB and SM Stations	2012	-	details
KO	AKDN	35.2970 N	32.9683 E	85	Yes	No		Akdeniz-KKKTC	Kandilli Observatory BB and SM Stations	2008	-	details

Station Detail

Select by Network
Network KO
Station TVSB
Description
Station Details
Station Pictures
Housing & Building details
Instrumentation History
Borehole

Metadata files EIDA

Dataless SEED
SC3 Inventory XML
FDSN Station XML

Description EIDA data & Ownership

Network	KO	**Latitude [°]**	39.449700 N	**Country**	-
Station Code	TVSB	**Longitude [°]**	29.461500 E	**Station Name**	Tavsanli-KUTAHYA
Affiliation	Kandilli Observatory BB and SM Stations	**Elevation [m]**	1090	**Description**	Tavsanli-KUTAHYA
Shared / Restric.	Yes / No	**Owner Name**	-	**Owner Phone**	-
Start	2009	**Owner Department**	-	**Owner Email**	-
End	-	**Owner Agency**	-	**Owner Address**	-

Station Details Morphology, Ground type, Geology, etc.

Geological Unit -
Morphology Class -
Classes: T1, T2, T3, T4; *based on the Italian building code*

Morphology Description -
Ground type EC8 -
EC8 types: A,B,C,D,E,S_1,S_2; *more info here*

Groundwater Depth [m] -
Vs 30 [m/s] -
f_0 [Hz] -
Fundamental frequency at the site

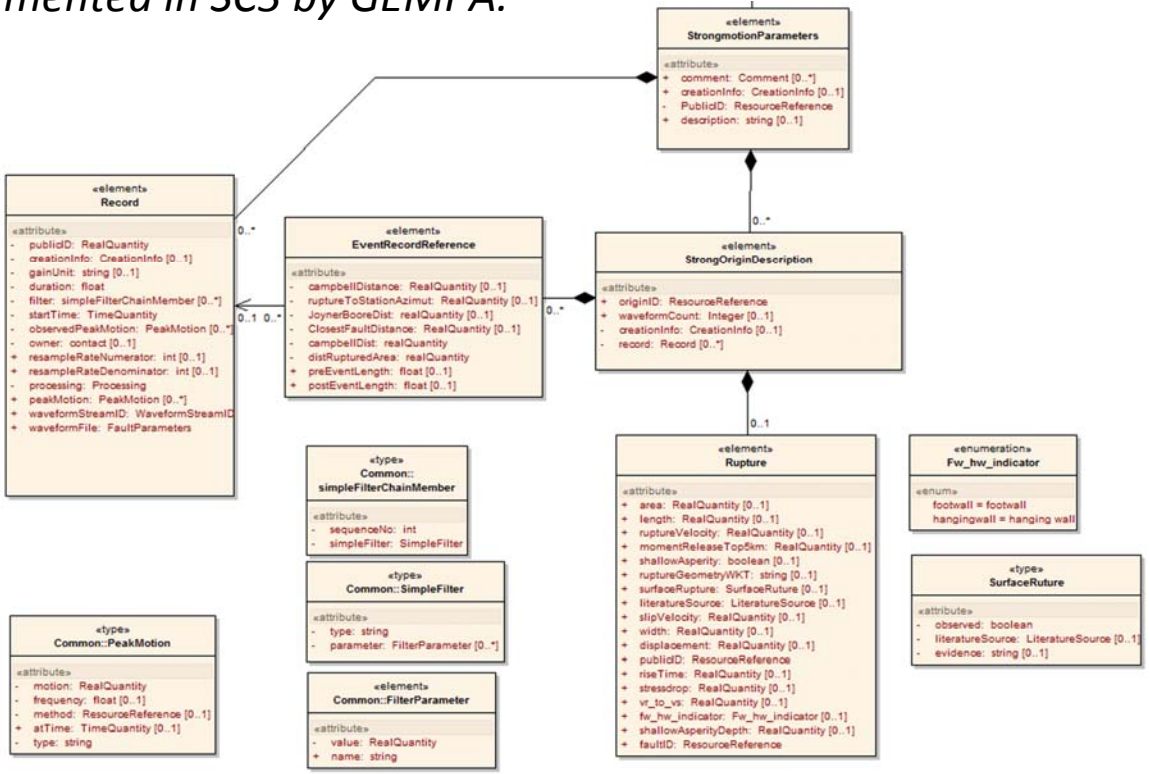
Amp(f_0) -
H/V amplitude at f_0

Basin Flag -
Bedrock Depth [m] -
Depth to the engineering bedrock (with Vs ~ 800 m/s)

RRSM data model (NERA NA3)

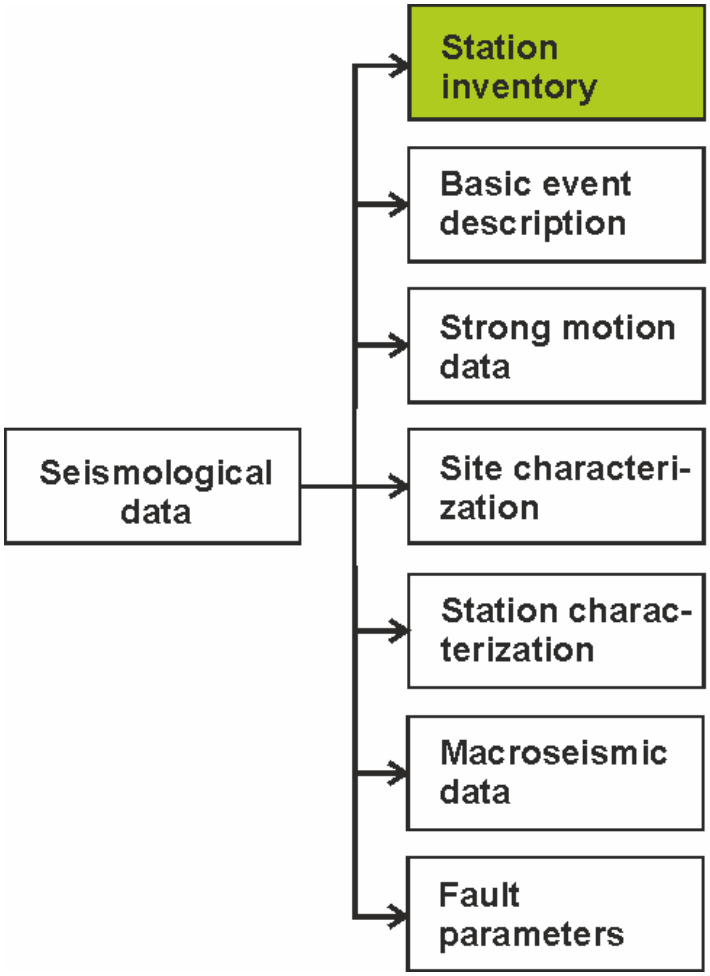
*ETH designed and developed a **data model** for strong-motion data and metadata for the SC3 database and a automatic processing module 'scwfparam' [discussed with NA3 participants];*

*Module '**scwfparam**' implemented in SC3 by GEMPA.*



Online documentation: http://www.seiscomp3.org/wiki/doc/applications/scwfparam

RRSM datamodel



SC3 datamodel (e.g. stations, affiliation, response)

*Filled trough automated **EIDA** synchronization*

QuakeML & SC3 data model (origin, picks, mag)

*Filled by **SC3** real-time processing*

NA3 extended event and strong motion info (e.g peak values). *Filled by '**scwfparam**' real-time processing.*

To be filled in by **network/station operator** through webinterface (part of StationBook)

Auxilliary information types (e.g. NA3 description of triggered fault)

SEED

- extended support for dataless SEED or end-of-life ? (documentation, builder s/w, …)
- mini-SEED limitations/modifications:
  - expanded network field (3 characters ? network blockette ?)
  - byte order, record length, encoding, microseconds, timing quality in fixed record header (Tim)
  - clean up data record blockettes ?

    100 – sample rate blockette (actual sample rate)
    200 – generic event detection blockette
    201 – Murdock event detection blockette
    202 – Log-Z event detection blockette
    300 – step calibration blockette
    310 – sine calibration blockette
    320 – pseudo-random calibration blockette
    390 – generic calibration blockette
    395 – calibration abort blockette
    400 – beam blockette
    405 – beam delay blockette
    500 – timing blockette (time down to microsecond)
    1000 – data only blockette (encoding key)
    1001 – data extension blockette (timing quality 0 – 100 %; vendor specific)
    2000 – variable length opaque data blockette

  - versioning blockette (or by use of reserved byte ?)
  - forward compatibility mapping, documented procedure (Tim)

Fixed section of data record (48 bytes)

	Note	Field name	Type	Length	Mask or Flags
	1	Sequence number	D	6	"######"
V2.4 -	2	Data header/quality indicator ("D"|"R"|"Q"|"M")	A	1	
	3	Reserved byte ("Δ")	A	1	
	4	Station identifier code	A	5	[UN]
	5	Location identifier	A	2	[UN]
	6	Channel identifier	A	3	[UN]
V2.3 -	7	Network Code	A	2	[ULN]
	8	Record start time	B	10	
	9	Number of samples	B	2	
	10	Sample rate factor	B	2	
	11	Sample rate multiplier	B	2	
	12	Activity flags	B	1	
	13	I/O and clock flags	B	1	
	14	Data quality flags	B	1	
	15	Number of blockettes that follow	B	1	
	16	Time correction	B	4	
	17	Beginning of data	B	2	
	18	First blockette	B	2	

Representing negative normalization factor (SEED, StationXML)

In cases where G(f) corresponds to an analog-type stage, a Poles and Zeros type response Blockette [53] is normally used to specify this stage. In this case, R(f) is expressed in this form:

$$R(f) = A_0 H_p(s) \qquad (5)$$

where $s = i\,2\,\pi\,f$ or $s = i\,f$ ($i = \sqrt{-1}$) as specified below equation (6) and $H_p(s)$ represents the transfer function ratio of polynomials specified by their roots, as in equation (6). For proper normalization, we chose A_0 such that $|R(f_s)| = 1.0$; that is $A_0 = 1/|H_p(s_s)|$, where $s_s = i 2\pi f_s$ $\frac{\text{rad}}{\text{sec}}$ or $s_s = i\,f_s$ (depending on whether we have represented the poles and zeros of Hp in terms of radians per second or Hz).

In cases where G(f) corresponds to an analog-type stage and the coefficient representation is used, as in equation (7), then the coefficients a_j and b_j of $H_c(s)$ are chosen such that $H_c(s_s)|=1.0$, where $s_s = i\,2\,\pi\,f_s$ or $s_s = i\,f_s$ Hz.

When G(f) corresponds to a digital-type stage and is represented with poles and zeros, as is usually the case with IIR filters (those with feedback), we again chose $A_0 = 1/|H_p(z_s)|$ where $H_p(z)$ is defined as the ratio of polynomials in equation (11), and $z_s = e^{2\pi i f_s \Delta t}$, where Δt is the sample interval and f_s is specified in the stage description.

- A_0 always positive by this SEED definition **but** poles and zero with negative A_0 are not uncommon

Solution:

- determine poles and zeros with positive A_0 (ideal but not straightforward)
- attribute the minus sign to the gain

Finally, when G(f) specifies a digital-type filter and is represented with coefficients, as is usually the case with FIR filters (those without feedback), the coefficients b_n of $\boldsymbol{H}_c(z)$ in equation (9) are chosen such that $|\boldsymbol{H}_c(z_s)| = 1.0$, where z_s is defined as in the previous paragraph.

This normalization works for stages 1 through K. If Blockette [58] has a stage number of 0, SEED assumes that the sensitivity S_d given in field 4 of Blockette [58] applies to the system as a whole, at the frequency f_s given in field 5 of Blockette [58]. Note that f_s should, if possible, be equal to the normalization frequency f_n given in any of stages 1 through K. In fact, within any stage, f_s should be equal to f_n. If no other stages are specified, SEED programs should conclude that this is our total knowledge of the system response. If we specify other stages, the stage-zero sensitivity will serve as a check on the sensitivity we can arrive at by multiplying together the responses $G_1, \ldots, G_K$. In this case, the stage-zero sensitivity is not multiplied together with the gains of the other stages. Rather, the stage-zero sensitivity should be equal to the product of the gains of the other stages at frequency $f_s = f_n$. If we have not used the same frequencies, f_s and f_n, for all stages 1 through K, then we can only say that the product of the sensitivities for each stage may be approximately equal to the stage 0 sensitivity. Note that this idea is much more intuitive and easier to work with if f_s and f_n are the same for all stages.

A possible exception is when the stage is a low pass digital FIR filter. The stage sensitivity for a FIR stage may be stated at $f_s = 0$ Hz (DC) if the in-band ripple is less than say, 1 or 2%. The DC gain of an FIR filter is the sum of the coefficients and so is easy to calculate.

overall sensitivity $S_d = G_1 * G_2 * G_3 \ldots$ so $S_d < 0$ when one of $G_i < 0$

The fullest possible specification will utilize Blockettes [53] and [58]. Blockette [53] will specify N zeros, $r_1, r_2, \ldots, r_N$ M poles $p_1, p_2, \ldots, p_M$, a normalization factor A_0, and a reference frequency. The normalization factor A0 and the reference, or normalization, frequency are specified in fields 7 and 8 of Blockette[53], the normalization frequency is in Hertz. Blockette [58] will specify a scaling factor S_d. Then at any frequency f (in Hz), the response is:

$$G(f) = S_d A_0 \frac{\prod_{n=1}^{N} (s - r_n)}{\prod_{m=1}^{M} (s - p_m)} = Sd\ A0\ Hp\ (s) \qquad (6)$$

where s = i*2pi*f if the transfer function is in terms of omega (radians/second), and s = i*f if the transfer function is in terms of frequency in Hertz. In both cases f is the frequency in Hertz.

Using two multiplicative coefficients, A_0 and S_d, in the equation above appears to be redundant, but we suggest that you partition the response by choosing A_0 so that the modulus of A_0 times the modulus of the ratio of polynomials equals 1.0 at the normalizing frequency f_n (also specified in Blockette [53]); the S_d specified in Blockette [58] is then the stage gain at that frequency, so $| G(f_n) | = S_d$. This division allows Blockette [53] to remain the same for many systems, with

this imples S_d always > 0 and is therefore confusing

re-write: $G(f_N) = S_d$

proposal: attribute the minus sign of A_0 to the gain

QC metrics

- QC or waveform parameterization (data metrics ?)
- levels of “QC parameters”
    - data record header flags (mini-SEED)
    - derived from sample values
    - derived ‘products’ using metadata and external source information (e.g. PSD, SNR, timing ….)
    - SOH

- developments IRIS DMC (Mustang)
- developments ORFEUS EIDA
- similarities / commonalities in algorithms and metrics -> FDSN standard

EIDA metrics	MUSTANG metrics	Description		Suggested FDSN standard
ms_amplifier_saturation	amplifier_saturation	# of records with this bit set in 'dq_flags'		ms_amplifier_saturation
ms_digital_filter_charging	digital_filter_charging	# of records with this bit set in 'dq_flags'		ms_digital_filter_charging
ms_digitizer_clipping	digitizer_clipping	# of records with this bit set in 'dq_flags'		ms_digitizer_clipping
ms_glitches	glitches	# of records with this bit set in 'dq_flags'		ms_glitches
ms_missing_padded_data	missing_padded_data	# of records with this bit set in 'dq_flags'		ms_missing_padded_data
ms_spikes	spikes	# of records with this bit set in 'dq_flags'		ms_spikes
ms_suspect_time_tag	suspect_time_tag	# of records with this bit set in 'dq_flags'		ms_suspect_time_tag
ms_telemetry_sync_error	telemetry_sync_error	# of records with this bit set in 'dq_flags'		ms_telemetry_sync_error
ms_calibration_signal	calibration_signal	# of records with this bit set in 'act_flags'		ms_calibration_signal
ms_event_begin	event_begin	# of records with this bit set in 'act_flags'		ms_event_begin
ms_event_end	event_end	# of records with this bit set in 'act_flags'		ms_event_end
ms_event_in_progress	event_in_progress	# of records with this bit set in 'act_flags'		ms_event_in_progress
ms_timing_correction	timing_correction	# of records with this bit set in 'act_flags'		ms_timin_corrrection
ms_clock_locked	clock_locked	# of records with this bit set in 'io_flags'		ms_clock_locked

All metrics are per time window (default: 24 hours, from day1 *00:00:00.0* to day2 *00:00:00.0*) – including first and last record containing the day boundary

Many of these bits are specific for mini-SEED and often of limited usefulness as they are handled differently by manufacturers and/or data centers - or not at all - and can thus <u>not</u> provide consistent information => pre-pend metrics name by 'ms_'

EIDA metrics	MUSTANG metrics	Description	Differences	Suggested FDSN standard
num_gaps	num_gaps	number of gaps	Δt $\Delta t + \varepsilon$ (predefined tolerance)	num_gaps
num_overlaps	num_overlaps	number of overlaps	Δt $\Delta t + \varepsilon$ (predefined tolerance	num_overlaps
gaps_len		total length of gaps	-	gaps_len
overlaps_len		total length of overlaps		overlaps_len
	max_gap	largest gap length in deconds		max_gap
	max_overlap	largest overlap length in seconds		max_overlap
percent_availability	percent_availability	percentage of available data	-	percent_availability
sample_max	sample_max	maximum sample value	-	sample_max
sample_min	sample_min	minimum sample value	-	sample_min
sample_mean	sample_mean	average sample value	-	sample_mean
sample_rms	sample_rms	root mean square sample value	-	sample_rms
sample_stdev		standard deviation sample value	-	(sample_stdev)
		mode ?		

EIDA metrics	MUSTANG metrics	Description	Diff.	Suggested FDSN standard
ms_timing_quality_mean	timing_quality	Mean percentage of timing quality based on blockette 1001	-	ms_timing_quality_mean
ms_timing_quality_max	-	Max. timing quality (bl. 1001)		ms_timing_quality_max
ms_timing_quality_min	-	Min. timing quality (bl. 1001)		ms_timing_quality_min
sample_rate_dev		Indicates a difference between sample rate in data header and in blockette 100 (actual sample rate)		ms_sample_rate_dev
sample_rate		sample rate in data header		
record_len		record length of first record		
number_records		total number of records		
size_in_bytes		total size of data in bytes		

MUSTANG metrics	Description	
asl_coherence	coherence between co-located BB sensors	
channel_up_time	duration of continuous segments of data (gaps > 1 sec allowed, duration < 30 sec omited)	
cross_talk	cross-corelate 2 channels same station for event > 5.5	
data_latency	data latency	
dc_offset	long term dc shift detector	
dead_channel_exp		
feed_latency		
max_stalta		
orientation_check		
pct_above_nhnm		
psd		
pressure_effects		
sample_median		
sample_snr		
spikes		
station_completeness		
suspect_time_tag		
total_latency		
transfer_function		