Information Model for LMAP

draft-burbridge-lmap-information-model-01

T. Burbridge, BT
P. Eardley, BT
M. Bagnulo, UC3M
J. Schönwälder, Jacobs University

Motivation / Example

Run the 'download speed test' with the test server at the end user's first IP hop in the network; if the end user is active then delay the test and re-try 1 minute later, with up to 3 re-tries; repeat every hour at xx.05 + Unif[0,180] seconds.

Report results once a day in a batch at 4am + Unif[0,180] seconds; if the end user is active then delay the report 5 minutes.

Copied from `draft-ietf-lmap-framework-01`

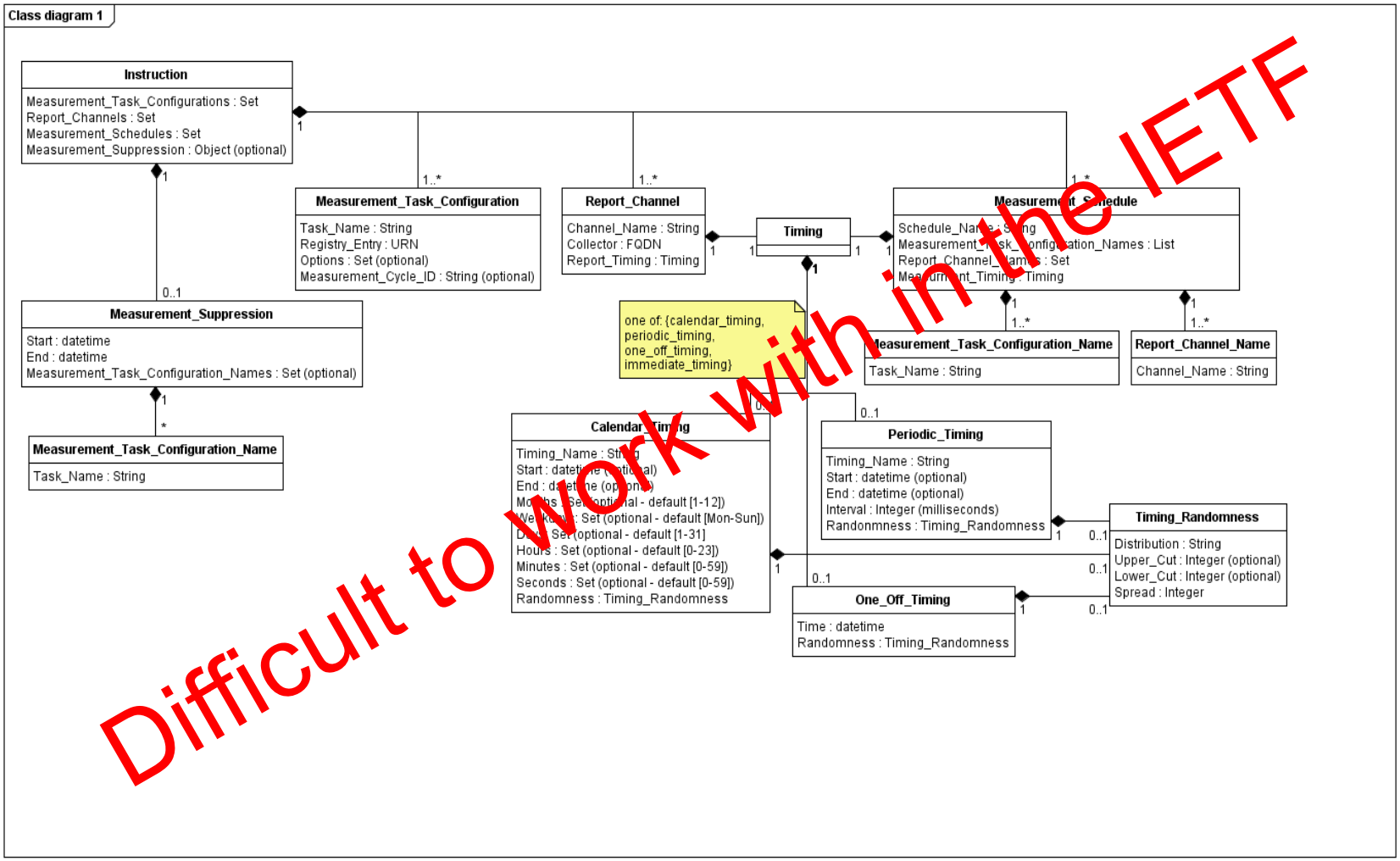
Information Model Sections

- Breaking the information model into *sections*:
  - Information needed for different functions
    - test parameters and test scheduling
    - report parameters and report scheduling
    - logging and status information
    - ...
  - Information in different sections is likely updated at different times, at different rates, etc.

Information Model Sections

Pre-Configuration	Minimal set of information necessary for an MA to securely contact an initial Controller
Configuration	Information configured by the Controller pertaining to Controller communication or general MA settings such as MA and Group ID
Instruction	Configuration by the Controller of what Measurement Tasks to perform, when to perform them, and where/when to report the results
Logging	Information transmitted back to the Controller with configuration or instruction errors and general failure notices
Status	Information available to be fetched by the Controller such as the Measurement Tasks supported by the MA
Reporting	Information sent to the Collector regarding the Measurement Task results including MA context and Task Configuration

Instruction Information



Instruction

```
ma-instruction {
  ma-tasks set of {
    ma-task-name     string;
    ma-task-registry urn;
    ma-task-options  set of string;
    ma-task-cycle-id string;                    /* optional */
  }
  ma-report set of ma-channel;
  ma-schedules set of {
    ma-schedule-name            string;
    ma-schedule-tasks           list of strings; /* task names */
    ma-schedule-report-channels set of string;   /* channel names */
    ma-schedule-timing          ma-timing;
  }
  ma-suppression set of {
    ma-suppression-start datetime;
    ma-suppression-end   datetime;
    ma-suppression-tasks set of string;         /* optional */
  }
}
```

Channel

```
ma-channel {
   ma-channel-name          string;
   ma-channel-target        url;
   ma-channel-cert          x509;
   ma-channel-timing        ma-timing;
}
```

Timing

```
ma-timing {
  ma-timing-name string;
  one of {
    ma-periodic-timing;
    ma-calendar-timing;
    ma-oneoff-timing;
    ma-immediate-timing;
  }
}

ma-periodic-timing {
  ma-periodic-start      datetime;
  ma-periodic-end        datetime;
  ma-periodic-interval   integer;       /* unit ms */
  ma-periodic-randomness ma-randomness; /* optional */
}

ma-immediate-timing {
}

ma-oneoff-timing {
  ma-oneoff-time       datetime;
  ma-oneoff-randomness ma-randomness;   /* optional */
}
```

Timing (cont)

```
ma-calendar-timing {
  ma-calendar-start      datetime;                /* optional */
  ma-calendar-end        datetime;                /* optional */
  ma-calendar-months     set of integer (1-12); /* optional */
  ma-calendar-weekdays   set of enum;             /* optional */
  ma-calendar-days       set of integer (1-31); /* optional */
  ma-calendar-hours      set of integer (0-23); /* optional */
  ma-calendar-minutes    set of integer (0-59); /* optional */
  ma-calendar-seconds    set of integer (0-59); /* optional */
  ma-calendar-randomness ma-randomness;           /* optional */
}

ma-randomness {
  ma-rand-distribution string;
  ma-rand-upper-cut    integer;          /* optional */
  ma-rand-lower-cut    integer;          /* optional */
  ma-spread            integer;
}
```

Reporting

```
ma-report {
  ma-report-date          datetime;
  ma-report-ma-id         string;          /* optional */
  ma-report-group-id      string;          /* optional */
  ma-report-context       set of XXX;      /* optional */
  ma-report-tasks         set of {
    ma-report-task-config XXX;
    ma-report-headers       list of string;
    ma-report-results list of {                  // XXX set of?
      ma-report-time           datetime;
      ma-report-cross-traffic  integer;          /* optional */
      ma-report-values         list of data;
    }
  }
}
```

(Pre-)Configuration / Logging

```
ma-config {
  ma-id           uuid;
  ma-group-id     string;
  ma-report-id    boolean;
  ma-instruction  ma-channel;
  ma-status       ma-channel;
  ma-logging      ma-channel;
}

ma-preconfig {
  ma-mac     mac-address;
  ma-config  channel;
  ma-cert    certificate;        /* optional */
  ma-id      uuid;               /* optional */
  ma-passwd  string;             /* optional */
}

ma-logging {
  ma-log-time datetime;
  ma-log-event XXX;
}
```

Status

```
ma-status {
  ma-stat-ma-id    string;
  ma-stat-device   string;
  ma-stat-hardware string;                /* optional */
  ma-stat-firmware string;                /* optional */
  ma-stat-software string;                /* optional */

  ma-stat-last-measurement   datetime;
  ma-stat-last-report        datetime;
  ma-stat-last-instruction   datetime;
  ma-stat-last-configuration datetime;

  ma-stat-supported-measurements set of {
    ma-sup-measurement urn;
    ma-sup-version     string;
  }
```

Status (cont)

```
  ma-stat-interfaces set of {
    ma-if-name   string;
    ma-if-type   string;
    ma-if-speed  integer;                 /* unit mbps */
    ma-if-lladdr string;
    ma-if-ipaddr set of {
      ma-if-ipaddr-version enum;
      ma-if-ipaddr-addr    string;
    }
    ma-if-gateway set of {                /* optional */
      ma-if-gateway-version enum;
      ma-if-gateway-addr    string;
    }
    ma-if-dns-server set of {             /* optional */
      ma-if-dns-server-version enum;
      ma-if-dns-server-address string;
    }
  }
}
```