

Archetype Definition Language (ADL) - an Introduction

Slides prepared by Thomas Beale

Ocean Informatics, Australia

www.OceanInformatics.biz



Presented by David Lloyd

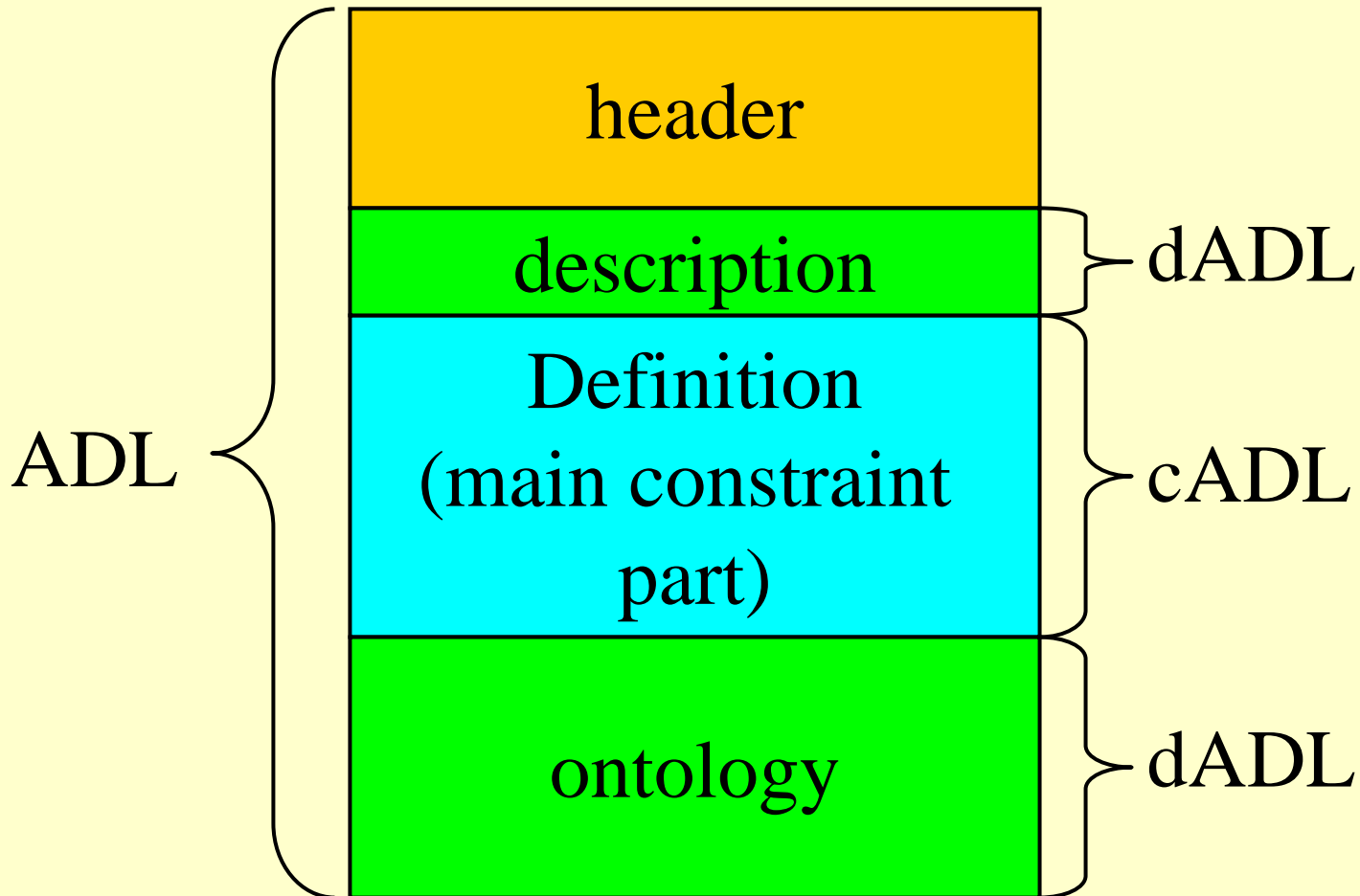
Chime at UCL

www.chime.ucl.ac.uk

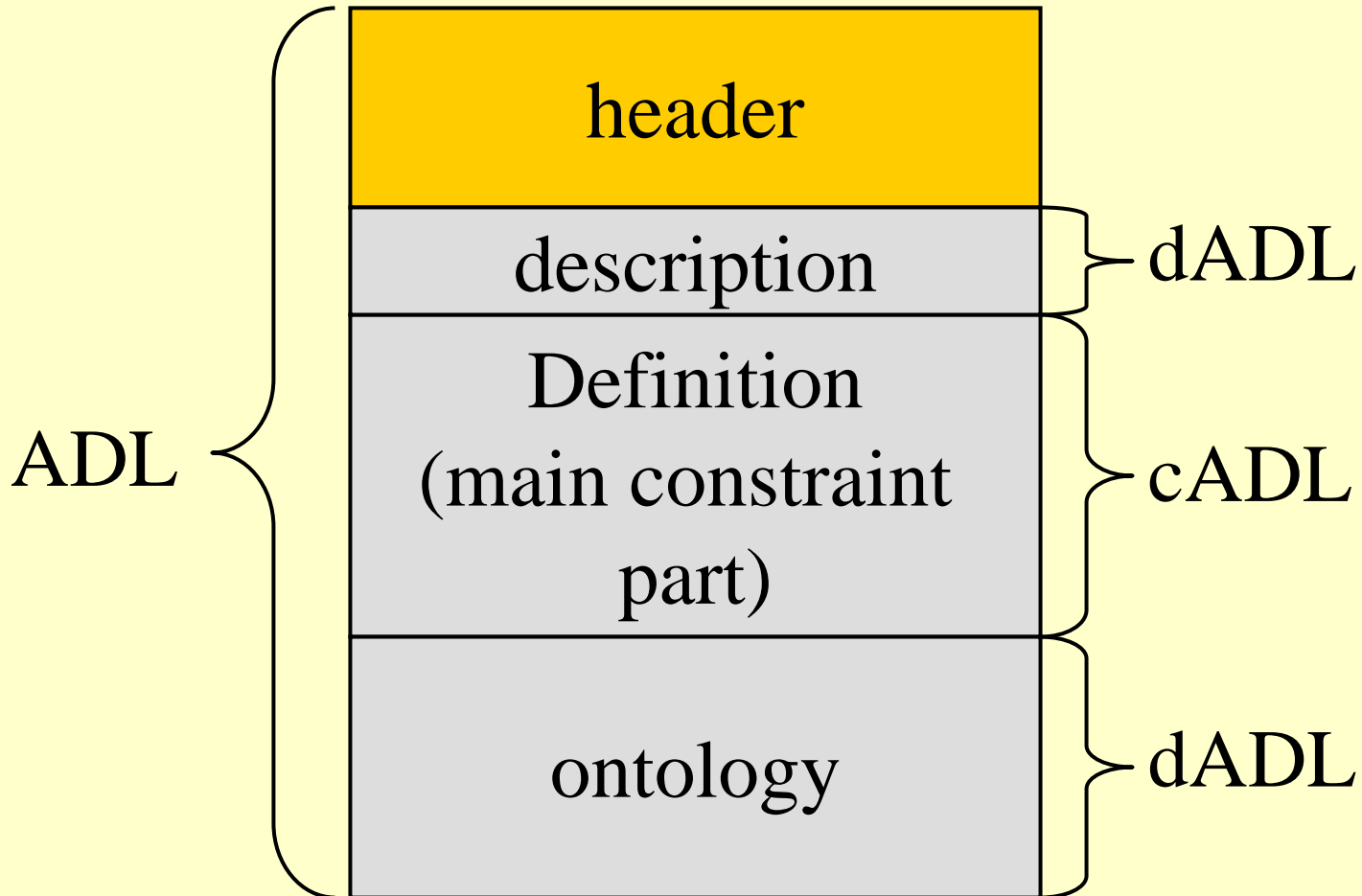
openEHR Foundation – <http://www.openEHR.org>

openEHR

ADL Document Structure



ADL Header



Header

archetype

`some.archetype.id`

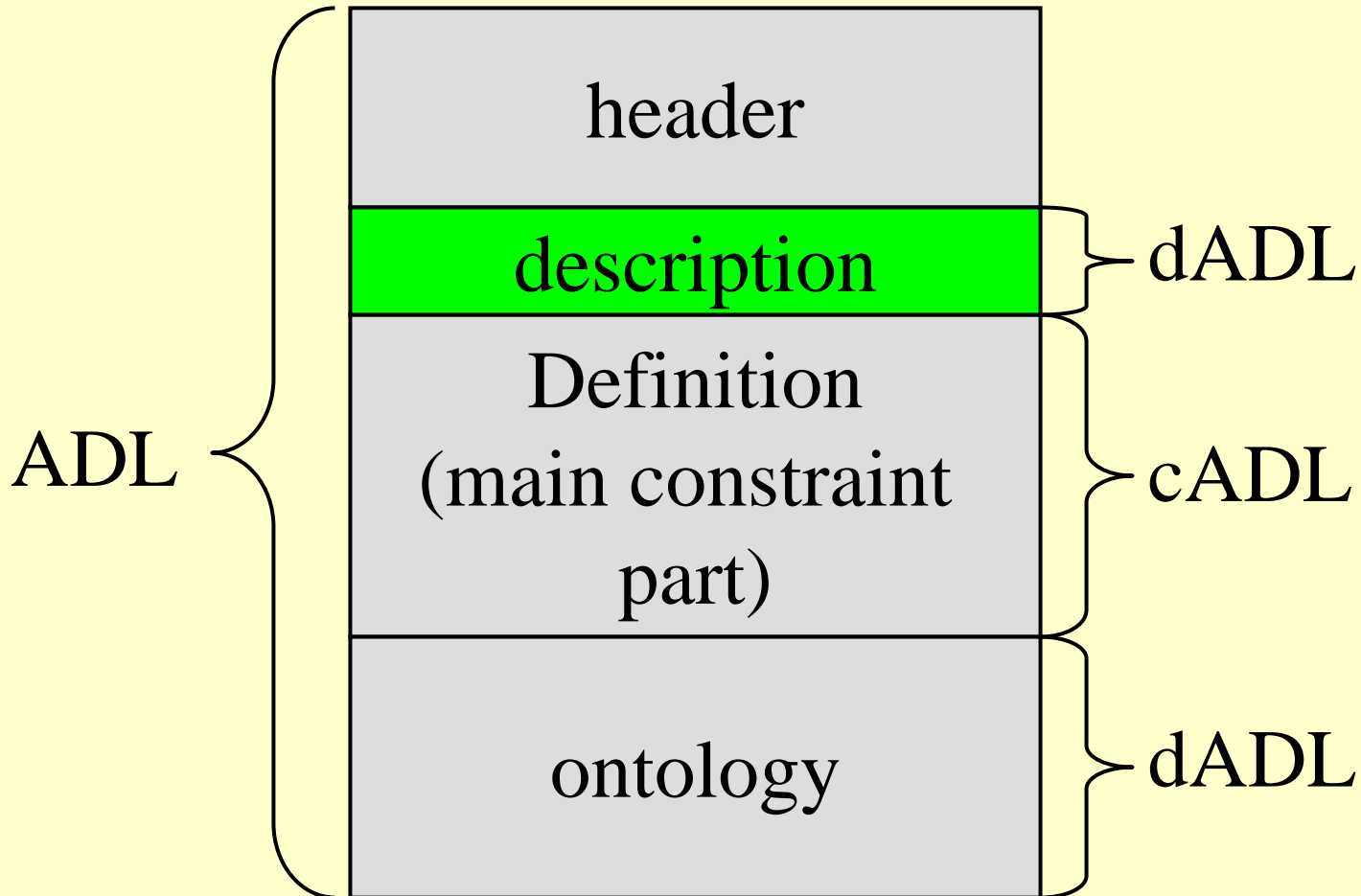
specialises

`some.parent_arch.id`

concept

`[at0000] -- term`

ADL Description



Description (the meta-data)

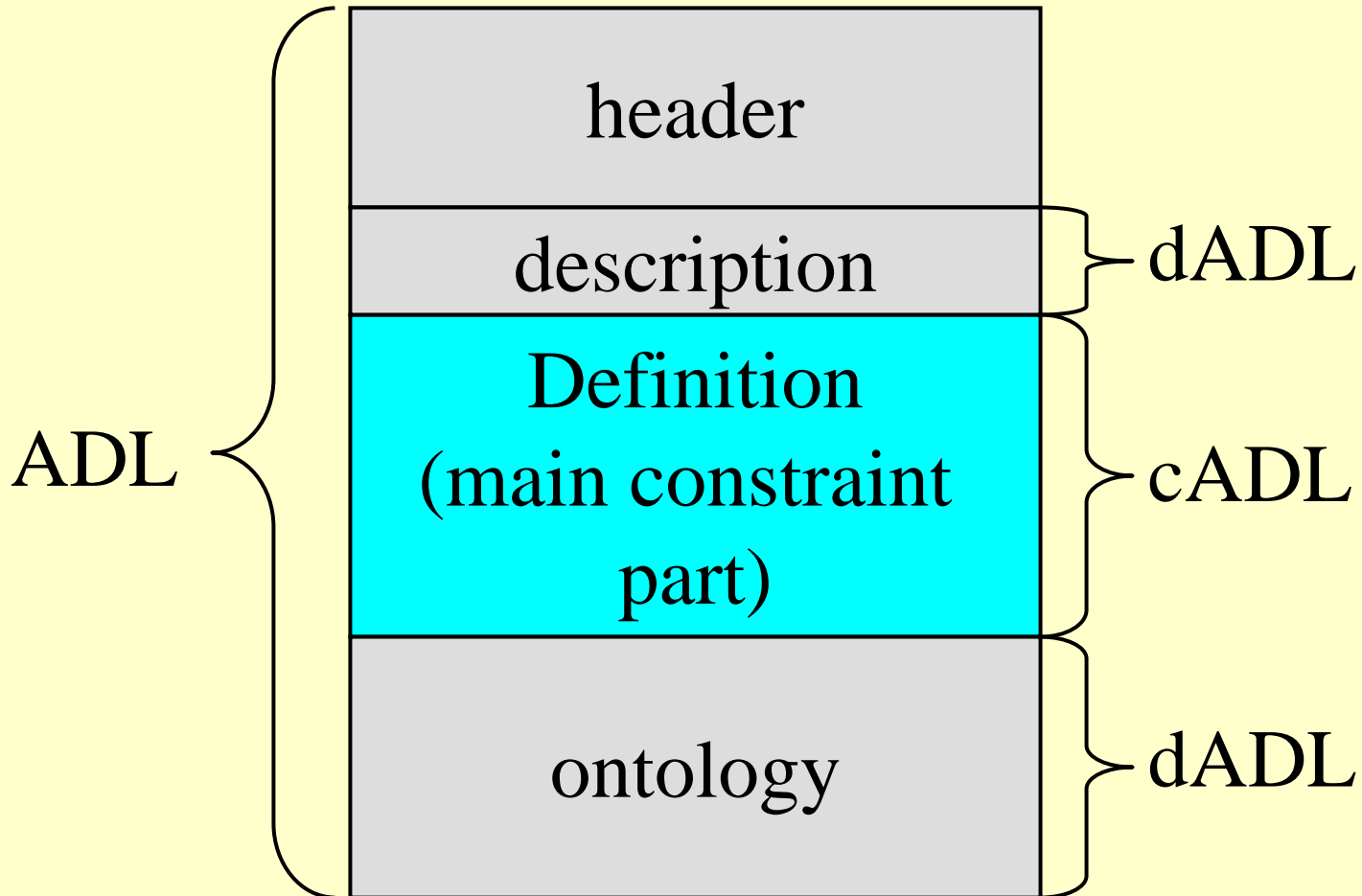
description

```
author = <"Sam Heard <s.heard@littlerock.com">>
submission = <
  organisation = <"WHO">
  date = <2003-12-20>
>
version = <"1.0">
status = <"draft">
revision = <"1.0">
description("en") = <
  purpose = <"Problem diagnosis evaluation">
  use = <"diagnosis description ..">
  misuse = <"differential diagnosis">
>
adl_version = <"1.0">
rights = <"© 2004 World Health Organisation">
```

Description continued...

- Actual model will conform to HL7 templates meta-data / Dublin Core, except that languages are supported properly
- ADL does not dictate the meta-data model

ADL Definition



Definition Section

- Main constraint definition of archetype
- Based on some reference model
- Syntax based on sets
- Completely compatible with UML object meta-model
- Convertible to various formats
- Uses OCL for invariants

Definition - overview

definition

```
TYPE_1 ∈ {  
  attr_1 ∈ {  
    TYPE_2 ∈ {  
      attr_a ∈ {yyyy-mm-??}  
      attr_b ∈ {|0.5..0.75|}  
    }  
  }  
}
```

- Each occurrence of $XXXX \in \{\text{constraint}\}$ is a specification of an *instance space* that the constrained data must fit into at runtime

Definition – node identifiers

```
TYPE_1[at0002] ∈ {  
  attr_1 ∈ {  
    TYPE_2[at0005] ∈ {  
      attr_a ∈ {YYYY-mm-??}  
      attr_b ∈ {|0.5..0.75|}  
    }  
  }  
}
```

- Node identifiers:
 - provide the domain meaning to each node
 - Are the basis of paths (multi-lingual & machine-processable)
 - Enable archetype nodes to be recorded in data
 - Defined in the ontology

Definition – existence

```
ELEMENT[at0002] ∈ {  
    value existence ∈ {0..1} ∈ {  
        QUANTITY ∈ {..}  
    }  
}
```

- Specify existence for attributes: 0..1, 1..1 (I.e. optional or mandatory)
- Defaults to 1..1

Definition – cardinality

```
LIST[at0002] ∈ {  
  items cardinality ∈ {2..*} ∈ {  
    ELEMENT[at0005] ∈ {..} -- systolic  
    ELEMENT[at0006] ∈ {..} -- diastolic  
    ELEMENT[at0099] ∈ {..} -- any  
  }  
}
```

- Specify cardinality to indicate container attributes
- Existence possible as well: means – does container exist at all (even empty)?

Definition – occurrences

```
LIST[at0002] ∈ {  
  items cardinality ∈ {2..*} ∈ {  
    ELEMENT[at0005] occurrences ∈ {1..1} ∈ {...}  
    ELEMENT[at0006] occurrences ∈ {0..1} ∈ {...}  
    ELEMENT[at0099] occurrences ∈ {0..*} ∈ {...}  
  }  
}
```

- Occurrences indicates how many times a data element conforming to an archetype block may occur
- Default: 1..1

Definition – basic leaf constraints

```
TEST[at0002] ∈ {  
  string_attr1 ∈ {"something"}  
  string_attr2 ∈ {/this|that|something else/  
  string_attr3 ∈ {/cardio.*/  
  
  string_attr4 ∈ [ac0045] -- see ontology  
  
  boolean_attr1 ∈ {True}  
  boolean_attr2 ∈ {False}  
  boolean_attr3 ∈ {True, False}  
}
```

Definition – basic leaf constraints

```
TEST[at0002] ∈ {  
    integer_attr1 ∈ {55} -- treated as an interval  
    integer_attr2 ∈ {55, 75, 100}  
    integer_attr3 ∈ { |0..100| }  
    integer_attr4 ∈ { |>= 10| }  
  
    real_attr1 ∈ {0.0}  
    real_attr2 ∈ {1.0, 2.0, 3.0}  
    real_attr3 ∈ { |0.0..100.0| }  
    real_attr4 ∈ { |>= 10.0| }  
    real_attr5 ∈ { |-10.0..-5.0| }  
}
```

Definition – date/time leaf constraints

```
TEST[at0002] ∈ {  
    date_attr1 ∈ {yyyy-mm-dd}  
    date_attr2 ∈ {yyyy-??-??}  
    date_attr3 ∈ {yyyy-mm-??}  
    date_attr4 ∈ {yyyy-??-XX}  
    date_attr5 ∈ {1983-12-25}  
    date_attr6 ∈ {2000-01-01}  
  
    time_attr1 ∈ {hh:mm:ss}  
    time_attr2 ∈ {hh:mm:XX}  
    time_attr3 ∈ {hh:?:XX}  
    time_attr4 ∈ {hh:?:??}  
    time_attr5 ∈ {22:00:05.0}  
}
```

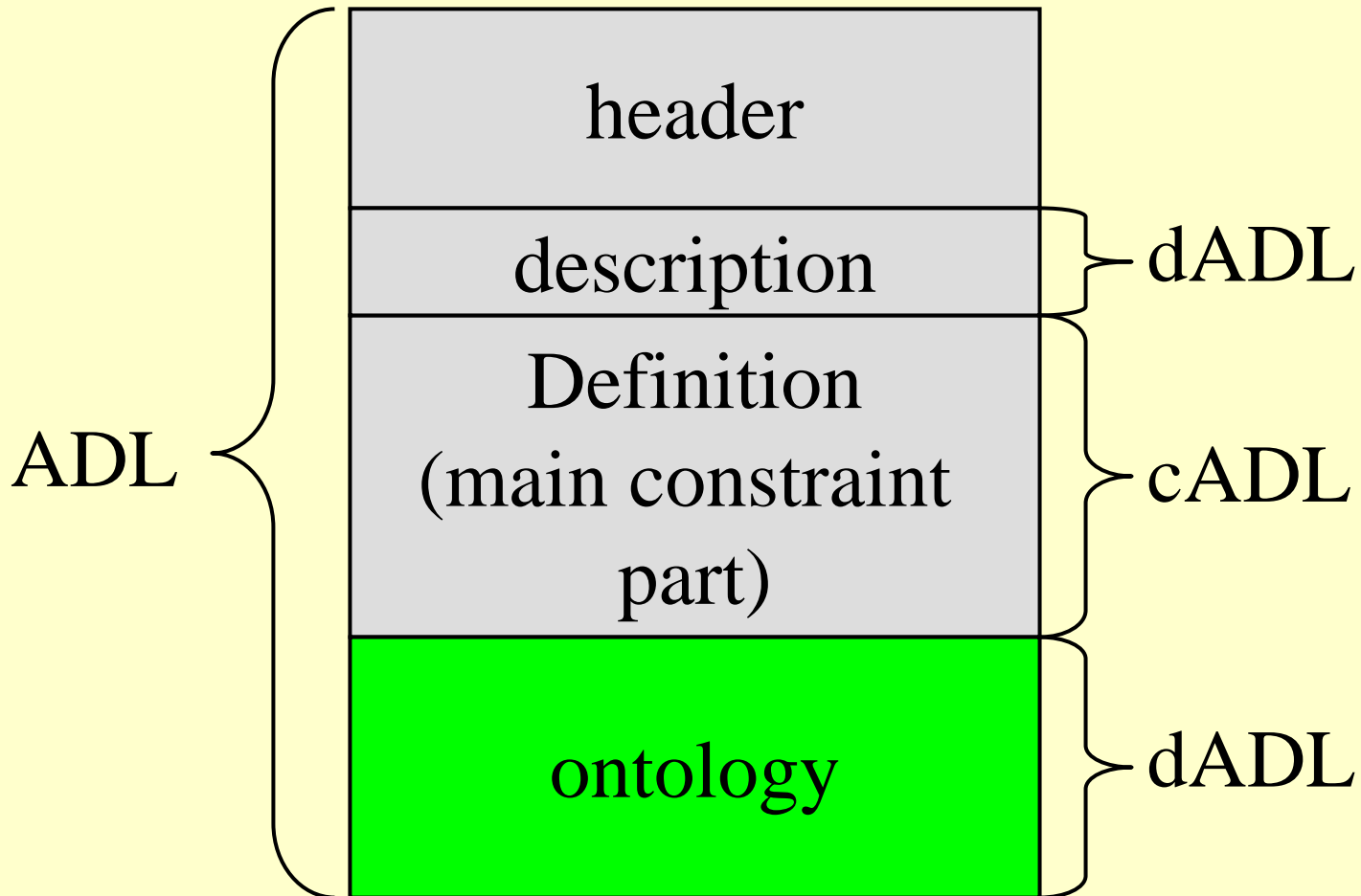
Definition – date/time leaf constraints

```
TEST[at0002] ∈ {  
    date_time_attr1 ∈ {yyyy-mm-dd hh:mm:ss}  
    date_time_attr2 ∈ {yyyy-mm-dd hh:mm:??}  
    date_time_attr3 ∈ {yyyy-mm-dd hh:mm:XX}  
    date_time_attr4 ∈ {yyyy-mm-dd hh:?:XX}  
    date_time_attr5 ∈ {yyyy-?-?-? ??:?:??}  
    date_time_attr6 ∈ {1983-12-25 22:00:05.0}  
  
    duration_attr1 ∈ {P0s}  
    duration_attr2 ∈ {P1d}  
    duration_attr3 ∈ {P2h5m}  
    duration_attr4 ∈ { |P1h55m..P2h5m| }  
    duration_attr5 ∈ { |<= P1h| }  
}
```

Definition – invariants

```
TEST[at0002] ∈ {  
  attr1 ∈ {  
    QUANTITY ∈ {  
      value ∈ {0..100}  
    }  
  }  
  attr2 ∈ {  
    QUANTITY ∈ {  
      value ∈ {0..100}  
    }  
  }  
  invariant  
    attr1/value >= attr2/value  
}
```

ADL Ontology



Ontology Section

- Contains local term definitions and bindings
- Managable in size
- Translations can be added without affecting main constraint definition
- Cost-effective to translate (cf all of snomed)
- Optional bindings, but not necessary

Ontology - overview

```
primary_language = <"en">
```

```
languages_available = <"en", "de">
```

```
terminologies_available = <"snomed-ct", "loinc">
```

```
term_definitions("en") = <...>
```

```
constraint_definitions("en") = <...>
```

```
term_binding("en") = <...>
```

```
constraint_binding("en") = <...>
```

- Archetype authored in one language
- Translations have to be with respect to primary language (basis of translation)
- Bindings to multiple terminologies supported

Ontology – term definitions

```
term_definitions("en") = < -- english
  items("at3121") = <
    text = <"Localised">
    description = <"Extent">
  >
  items("at3122") = <...>
>
term_definitions("tr") = < -- turkish
  items("at3121") = <
    text = <"Lokelize">
    description = <"Yayýlým">
  >
  items("at3122") = <...>
>
```

Ontology – constraint definitions

```
constraint_definitions("en") = < -- english
  items("at0005") = <
  text = <"patient position">
  description = <"patient position during BP
measurement">
>
>
```

Ontology – term bindings

```
term_binding("loinc") = <
    items("at1000") = <[loinc::700-0]>
items("at1001") = <[loinc::718-7]>
items("at1002") = <[loinc::718-7]>
..
>
term_binding("snomed") = <
    items("at0005") = <[snomed::20093944]>
..
>
```

Ontology – constraint definitions

```
constraint_binding("snomed") = <
  items("ac0010") = <query("terminology",
    "terminology_id = snomed_ct;
    has_relation [102002] -- is-a
    with_target [246153002]")-- auto-immune disease
  >
```

- Connection between constraints and underlying ontologies
- Language of query not yet defined (others will define it)
- Typically only approximate
- Almost always partial coverage

Summary

- ADL Formalism is simple and consistent
- Minor additions for Templates nearly complete
- Tools
 - OS Parsers for ADL, cADL, dADL exist
 - OS GUI archetype editor nearly complete