

# A Glossary of Time Granularity Concepts

Claudio Bettini<sup>1</sup>, Curtis E. Dyreson<sup>2</sup>, William S. Evans<sup>3</sup>,  
Richard T. Snodgrass<sup>3</sup>, and X. Sean Wang<sup>4</sup>

- <sup>1</sup> Dipartimento di Scienze dell'Informazione, Università degli Studi di Milano,  
Milano, Italy  
[bettini@dsi.unimi.it](mailto:bettini@dsi.unimi.it),  
<http://mercurio.sm.dsi.unimi.it/~bettini/>
- <sup>2</sup> Department of Computer Science, James Cook University, Townsville, Australia  
[curtis@cs.jcu.edu.au](mailto:curtis@cs.jcu.edu.au),  
<http://www.cs.jcu.edu.au/~curtis/>
- <sup>3</sup> Department of Computer Science, University of Arizona, Tucson, Arizona, USA  
[{will,rts}@cs.arizona.edu](mailto:{will,rts}@cs.arizona.edu),  
<http://www.cs.arizona.edu/people/{will,rts}>
- <sup>4</sup> Department of Information and Software Systems Engineering, George Mason  
University, Fairfax, Virginia, USA  
[xywang@isse.gmu.edu](mailto:xywang@isse.gmu.edu),  
<http://www.isse.gmu.edu/faculty/xywang>

## 1 Introduction

This paper is an extension of the preceding glossary, but focussed on time granularity concepts. We use the same structure as in the previous glossary.

## 2 Glossary Entries

This glossary contains revised definitions of *calendar* and *granularity* that appeared in the preceding glossary.

### 2.1 Time Domain

A *time domain* is a pair  $(T; \leq)$  where  $T$  is a non-empty set of *time instants* and  $\leq$  is a total order on  $T$ .

**Explanation** The time domain is the set of primitive temporal entities used to define and interpret time-related concepts. The set is ordered by a relationship,  $\leq$ , on these entities. The expression  $t < t'$  is used instead of  $t \leq t' \wedge t \neq t'$  to identify strict order.

The choice for time domain is typically between dense and discrete:

A time domain is *dense* if it is an infinite set and for all  $t, t' \in T$  with  $t < t'$ , there exists  $t'' \in T$  such that  $t < t'' < t'$ .

A time domain is *discrete* if every element except the last element (if any) has an immediate successor, and every element except the first (if any) has an immediate predecessor.

In general, if the application imposes a fixed smallest granularity (section 2.2), such as **seconds**, then a discrete time domain is probably the appropriate choice. However, if one is interested in being able to represent arbitrarily finer granularities, a dense time domain (e.g., the rational numbers with the usual order) is required. Other choices, as, for example, a continuous structure like the real numbers with the usual order, are also possible. In temporal logics, non-linear structures are also considered as time domains, but in database applications the linearity assumption seems to be reasonable.

Specific applications may also need to impose left/right boundedness to the time domain. A time domain is *bounded* if it contains upper and/or lower bounds with respect to its order relationship. Formally (in the case of both lower and upper bounds), time domain  $T$  is bounded if there exist  $t', t'' \in T$  such that  $t' \leq t \leq t''$  for all  $t \in T$ . ( $t'$  is called *beginning* in the preceding glossary.)

**Example** Integers ( $\mathbf{Z}, \leq$ ), natural numbers ( $\mathbf{N}, \leq$ ), rational ( $\mathbf{Q}, \leq$ ), and real numbers ( $\mathbf{R}, \leq$ ) are all examples of time domains.

## 2.2 Granularity

A *granularity* is a mapping  $G$  from the integers (the *index set*) to subsets of the time domain such that:

- (1) if  $i < j$  and  $G(i)$  and  $G(j)$  are non-empty, then each element of  $G(i)$  is less than all elements of  $G(j)$ , and
- (2) if  $i < k < j$  and  $G(i)$  and  $G(j)$  are non-empty, then  $G(k)$  is non-empty.

**Explanation** The first condition states that granules (section 2.3) in a granularity do not overlap and that their index order is the same as their time domain order. The second condition states that the subset of the index set that maps to non-empty subsets of the time domain is contiguous.

While the time domain can be discrete, dense, or continuous, a granularity defines a countable set of granules; each granule is identified by an integer. The index set can thereby provide an “encoding” of the granularity in a computer.

Independently, there may be a “textual representation” of each non-empty granule, termed its *label*, that is used for input and output. This representation is generally a string that is more descriptive than the granule’s index. An associated mapping, the *label mapping*, defines for each label a unique corresponding index. This mapping can be quite complex, dealing with different languages and character sets, or can be omitted if integers are used directly to refer to granules. For example, “August 1997” and “September 1997” are two labels each referring to the set of time instants (a granule) corresponding to that month.

A granularity is *bounded* if there exist lower and upper bounds  $k_1$  and  $k_2$  in the index set such that  $G(i) = \emptyset$  for all  $i$  with  $i < k_1$  or  $k_2 < i$ .

**Example** The usual collections **days**, **months**, **weeks** and **years** are granularities. The granularity describing all years starting from 1800, can be defined as a mapping that takes an arbitrary index  $i$  to the subset of the time domain corresponding to the year 1800,  $i + 1$  to the one corresponding to the year 1801, and so on, with all indexes less than  $i$  mapped to the empty set. The years from 1996 to 2000 can also be represented as a granularity  $G$ , with  $G(1996)$  identifying the subset of the time domain corresponding to the year 1996,  $G(1997)$  to 1997 and so on, with  $G(i) = \emptyset$  for each  $i < 1996$  and  $i > 2000$ .

### 2.3 Granule

Each non-empty subset  $G(i)$  is called a *granule* of the granularity  $G$ .

**Explanation** Intuitively, a granule is a set of time instants perceived as a non-decomposable temporal entity when used to describe, in terms of the granularity  $G$ , a phenomena or, in general when used to timestamp a set of data. A granule can be composed of a single instant, a set of contiguous instants (time-interval), or even a set of non-contiguous instants. For example, the September 1997 business-month, defined as the collection of all the business-days in September 1997, can be used as a granule.

Two granules  $G(i)$  and  $G(j)$  are *contiguous* if there does not exist  $t \in T$  such that  $G(i) < t < G(i + 1)$ . (We say  $S < t$  if and only if  $\forall s \in S, s < t$  and  $t < S$  if and only if  $\forall s \in S, s > t$ .)

### 2.4 Origin

The *origin* of granularity  $G$  is a specially designated granule, e.g.,  $G(0)$ .

**Explanation** This is an implementation-oriented concept, it is often useful to specially designate one granule. For example, the origin of **UNIX-calendar-seconds** is **Jan 1 0:00:01 GMT 1970**.

### 2.5 Anchor

The *anchor* of  $G$  is the greatest lower bound of the set of time domain elements corresponding to the origin of  $G$ .

**Explanation** Intuitively, the anchor is the earliest time domain element in the origin. It serves to “anchor” the granularity with respect to the time domain, when that granularity is specified in terms of another granularity.

## 2.6 Image

The *image* of a granularity is the union of the granules in the granularity.

**Example** The image of **business-days-since-1990** is the set of time instants included in each granule representing a business-day, starting from the first one in 1990.

## 2.7 Extent

The *extent* of a granularity is the smallest interval of the time domain that contains the image of the granularity. Formally, it is the set  $\{t \in T \mid \exists a, b \in I, a \leq t \leq b\}$  where  $T$  is the time domain and  $I$  is the image of the granularity.

**Example** In time domain  $(\mathbf{Z}, \leq)$ , the extent of **business-days-since-1990** is  $\{t \mid t \geq k\}$  where  $k$  identifies the first instant of the time domain within the first business-day of 1990.

Note that many commonly used granularities (e.g., **days**, **months**, **years**) have their image equal to their extent, since each granule is formed by a set of contiguous elements of the time domain and each pair of contiguous indexes is mapped to contiguous granules.

## 2.8 Granularity Relationships

The following are relationships defined between granularities having the same time domain.

**Groups Into** A granularity  $G$  *groups into* a granularity  $H$ , denoted  $G \trianglelefteq H$ , if for each index  $j$  there exists a (possibly infinite) subset  $S$  of the integers such that  $H(j) = \bigcup_{i \in S} G(i)$ .

**Finer Than** A granularity  $G$  is *finer than* a granularity  $H$ , denoted  $G \preceq H$ , if for each index  $i$ , there exists an index  $j$  such that  $G(i) \subseteq H(j)$ .

If  $G \preceq H$  then  $H$  is *coarser than*  $G$  ( $H \succeq G$ ).

**Sub-granularity** A granularity  $G$  is a *sub-granularity* of  $H$ , denoted  $G \sqsubseteq H$ , if for each index  $i$ , there exists an index  $j$  such that  $G(i) = H(j)$ .

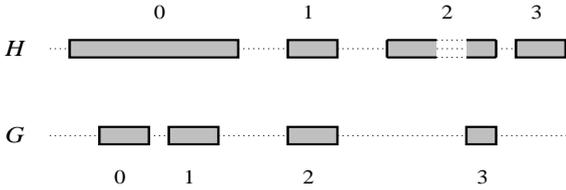
**Shift Equivalent** Granularities  $G$  and  $H$  are *shift equivalent*, denoted  $G \rightleftharpoons H$ , if there exists an integer  $k$  such that  $G(i) = H(i + k)$  for all  $i$  in the index set.

Note:  $G \rightleftharpoons H$  if and only if  $G \sqsubseteq H$  and  $H \sqsubseteq G$ .

**Partitions** A granularity  $G$  *partitions* a granularity  $H$  if  $G \trianglelefteq H$  and  $G \preceq H$ .

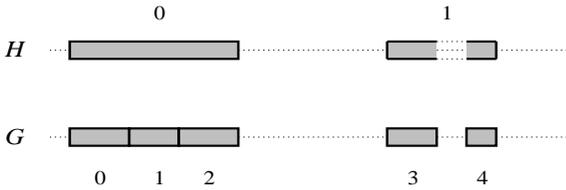


The next diagram depicts a *finer than* relationship.



$G \preceq H$  since every granule in  $G$  is a subset of some granule in  $H$ . For example, **business-days** is finer than **days**, and also finer than **weeks**.

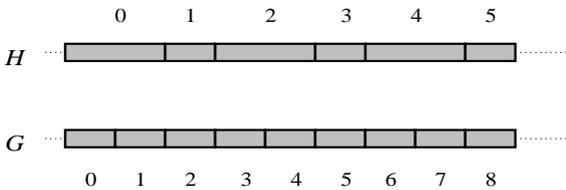
A relationship related to groups into, but more restrictive, is *partitions*. In the diagram below  $G$  partitions  $H$ .



Note the similarity to the groups into diagram given above, the only difference is that the “additional” granules in  $G$  that have no counterpart in  $H$  have been removed since the presence of either granule would violate the finer than relationship required for partitions. Many common (Gregorian calendar) granularities are related by partitions relationships, e.g., **days** partitions **weeks** and **months** partitions **years**.

The *groups periodically into* relationship is a special case of *groups into* characterized by a periodic repetition of the “grouping pattern” of granules of  $G$  into granules of  $H$ . Its definition may appear complicated but it is actually quite simple. Since  $G$  groups into  $H$ , any non-empty granule  $H(i)$  is the union of some granules of  $G$ ; for instance, assume it is the union of the granules  $G(a_1), G(a_2), \dots, G(a_k)$ . The periodicity property (condition 2 in the definition) ensures that the  $n^{\text{th}}$  granule after  $H(i)$ , i.e.,  $H(i+n)$ , if non-empty, is the union of  $G(a_1+m), G(a_2+m), \dots, G(a_k+m)$ . This results in a periodic “pattern” of the composition of  $n$  granules of  $H$  in terms of granules of  $G$ . The pattern repeats along the time domain by “shifting” each granule of  $H$  by  $m$  granules of  $G$ .

The following granularities  $G$  and  $H$  are such that  $G$  groups periodically into  $H$  with  $m$  and  $n$  being 3 and 2, respectively.



An instance of the “grouping pattern” is  $H(0) = G(0) \cup G(1)$  and  $H(1) = G(2)$ . This pattern repeats every 3 granules of  $G$ , with the first 2 forming one granule

of  $H$  and with the third forming the next one. For example,  $H(0 + 2) = G(0 + 3) \cup G(1 + 3)$  and  $H(1 + 2) = G(2 + 3)$ .

In general, this relationship guarantees that granularity  $H$  can be finitely described (in terms of granules of  $G$ ) providing the following information:

- (i) the finite sets  $S_0, \dots, S_{n-1}$  of indexes of  $G$  each one describing the composition of one of the  $n$  repeating non-empty granules of  $H$ ;
- (ii) the value of  $m$ ;
- (iii) the indexes of first and last non-empty granules in  $H$ , if their value is not infinite.

In the above example,  $H$  can be described by  $S_0 = \{0, 1\}$  and  $S_1 = \{2\}$  (two sets since  $n = 2$ ), and  $m = 3$ .

The description of arbitrary granules of  $H$  can be obtained by the following formula:

$$H(j) = \bigcup_{i \in S_{j \bmod n}} G(m * \lfloor j/N \rfloor + i)$$

This formula applies in general, provided that  $S_0, \dots, S_{n-1}$  are the sets of indexes of  $G$  describing  $H(0), \dots, H(n - 1)$ , respectively, and no granule of  $H$  is empty, but it can be easily adapted to the case with finite index for first and last non-empty granules.

Many common granularities are in this kind of relationship, for example, **months** groups periodically into **years** and **days** groups periodically into **years**. It is sometimes helpful to use  $m$  and  $n$  to more precisely specify how two granularities are related, e.g., **12 months** groups periodically into **1 year** and **14,697 days** groups periodically into **400 years**. Alternatively, the relationship can also be described, saying, for example, **years** is periodic (or 1-periodic) with respect to **months**, and **years** is periodic (or 400-periodic) with respect to **days**.

### 2.9 Bottom granularity

Given a granularity order relationship  $g\text{-rel}$  and a set of granularities having the same time domain, a granularity  $G$  in the set is a *bottom granularity* with respect to  $g\text{-rel}$ , if  $G \text{ } g\text{-rel} \text{ } H$  for each granularity  $H$  in the set.

**Example** Given the set of all granularities defined over the time domain  $(\mathbf{R}; \leq)$ , and the granularity relationship  $\preceq$  (finer than), the granularity corresponding to the *empty* mapping is the bottom granularity with respect to  $\preceq$ .

Given the set of all granularities defined over the time domain  $(\mathbf{Z}; \leq)$ , and the granularity relationship  $\trianglelefteq$  (groups into), the granularity mapping each index into the corresponding instant (same integer number as the index) is a bottom granularity with respect to  $\trianglelefteq$ .

An example of a set of granularities without a bottom (with respect to  $\preceq$  or  $\trianglelefteq$ ) is **{weeks, months}**. Both the *timestamp granularity* and the granularity of *chronons* (both are defined in the temporal glossary) are typically bottom granularities.

### 2.10 Granularity Lattice

A set of granularities, having the same time domain, forms a *granularity lattice* with respect to a granularity order relationship *g-rel* if for each pair of granularities in the set there exists a least upper bound and a greatest lower bound with respect to the relationship *g-rel*.

Note that this is the standard definition of lattice.

**Example** The set of granularities defined over the time domain  $(\mathbf{R}; \leq)$  such that no pair of granularities is shift equivalent is a granularity lattice with respect to  $\preceq$ .

### 2.11 Calendar

A *calendar* is a set of granularities over a single time domain that includes a bottom granularity with respect to  $\trianglelefteq$  (groups into).

**Explanation** Calendars are typically used to describe events or time-related properties over the same span of time using different granularities.

**Example** The Gregorian calendar comprises the granularities *days*, *months*, and *years*.

## 3 Quick Reference to Notation

We summarize the proposed notation in the following table.

$(T; \leq)$	Time Domain
$G$	Granularity
$G(i)$	Granule of $G$ with index $i$
$G \trianglelefteq H$	$G$ groups into $H$
$G \preceq H$	$G$ is finer than $H$
$G \sqsubseteq H$	$G$ is a <i>sub-granularity</i> of $H$
$G \tilde{\subseteq} H$	$G$ is covered by $H$
$G \dot{\sim} H$	$G$ is <i>shift equivalent</i> to $H$

### Acknowledgment

Bettini and Wang wish to thank Sushil Jajodia for his discussions with them on many terms that appear in this glossary.