INFO-455, Spring 2013, Class #1

Greetings …

Design problem scenario: A short brief for the "Factory Problem"

Consider a factory in which a large number of machines are in operation, generally 24 hours per day. Each of these machines has been instrumented so that an energy reading is generated every 15 minutes. These readings comprise what is called a time series, as, for example, the following:

```
…
(M1, 04/01/2013 – 12:05, 345.75)
(M2, 04/01/2013 – 12:05, 1254.33)
(M1, 04/01/2013 – 12:30, 401.5)
…
```

The machines generate data points, (**Mx, date-time, reading**), where **Mx** is the machine ID, **date-time** is the day and time that the reading was made, and **reading** is a measure of electricity use in kilowatt hours (kWh). Millions of readings might be generated in a year.

In addition, each machine has a model number, a location on the factory floor, and a set of records which indicate when a particular machine was inspected and repaired. In addition, different operators, that is, factory employees, run different machines during the 24hr day. When working, operators always log-in and log-out of a machine.

Consider: *What would be a suitable database schema for this setting?*

Questions:

1. What questions – if any – would you like to ask the client and/or other stakeholders?
2. What are the three important queries that the system needs to handle?
3. What should the data types be for Mx, date-time, and reading?
4. Draw an Entity-Relationship model of the above situation, showing all entities, relationships, and attributes. Please use 1-M and 1-1 relationships and show the participation constraints. In addition, please specify the data types of all your attributes. Provide a concise caption explaining your model and any assumptions that you have made.

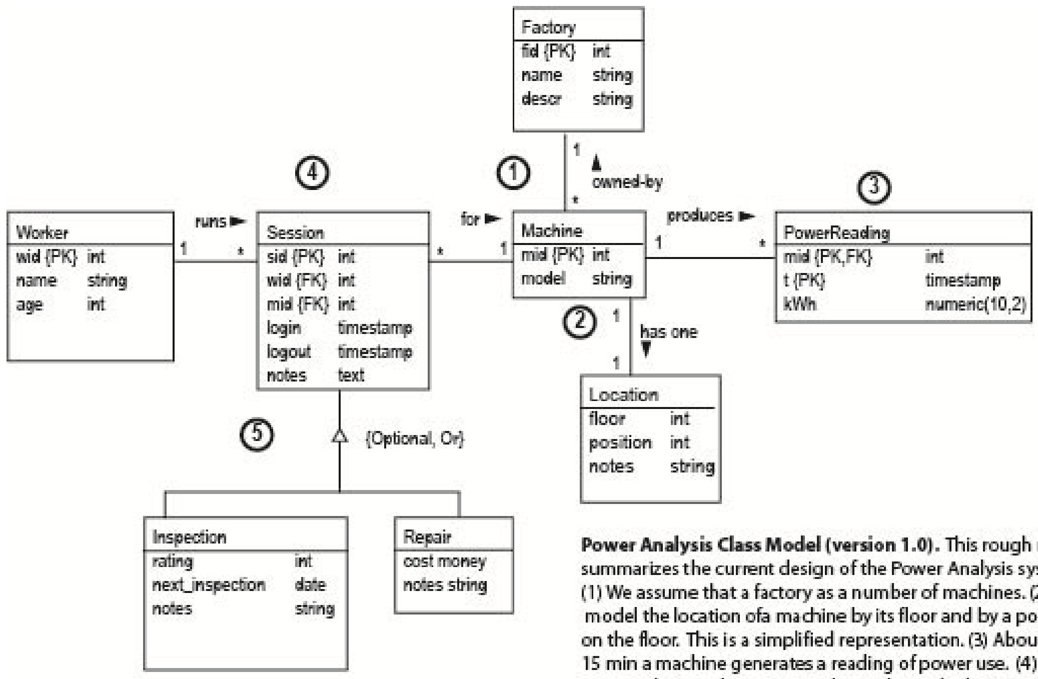
A rough Entity-Relationship model for the "Factory Application" problem that we have been discussing. The model is epxressed in UML (Universal Modeling Language) or, at least, a variant of UML which is used in our textbook (see, for example, http://cs-exhibitions.uni-klu.ac.at/index.php?id=435 which presents a brief timeline of modeling notations.)

In any case, modeling is extremely important because:

(1) The model provides a higher-level specification thats leads to directly to a specific physical implmentation;
(2) The model helps us think about the problem space and by modeling the domain we discover new questions to ask our clients;
(3) The model helps us communicate the members of a design team and other stakeholders on a project.

I've created this model using Adobe Illustrator which is a very good drawing tool. The file http://courses.washington.edu/info445/docs/uml_parts.ai (found on the website) provides a set of elements for using the UML notation. Projects involving fewer that 50 elements working by hand is almost always better than using a tool. In any case, please always follow the notational standards carefully, include a caption, and be as neat as possible.

Do you see any issues with this model? (*Hint*: There are always issues with ER models ☺)



Power Analysis Class Model (version 1.0). This rough model summarizes the current design of the Power Analysis system. (1) We assume that a factory as a number of machines. (2) We model the location ofa machine by its floor and by a position on the floor. This is a simplified representation. (3) About every 15 min a machine generates a reading of power use. (4) We assume that machines are run by workers who log-in and l og-out of the machines. (5) In addition, we assume that from time to time machines are inspected and repaired - similar to a session we shall assume that when workers inspect or repair a machine then log-in and log-out.

```
/*
 * A simple exploratory database
 */

/*
 * The system that we will be using plpgsql scripts
 */
create or replace language plpgsql;

/*
 * Create blank schema for holding this simple example
 */
DROP SCHEMA IF EXISTS fct CASCADE;
CREATE SCHEMA fct;

/*
 * Represent the factor -- allows application to be extended to multiple factories
 */
CREATE TABLE fct.factory (
   fid          SERIAL PRIMARY KEY,
   name         varchar(64) UNIQUE NOT NULL,
   descr        text
);

insert into fct.factory(name,descr) values ('factory #1', 'This is s test factory. (#1)');
insert into fct.factory(name,descr) values ('factory #2', 'This is s test factory. (#2)');

/* Question: Would this expression work? */
/* insert into fct.factory(descr) values ('A no name factory'); */

/*
 * Worker. Note: Some type checking of values are illustrated
 */
CREATE TABLE fct.worker (
   wid          SERIAL PRIMARY KEY,
   /* Do not allow numbers to be part of the name -- only allow letters, space and period*/
   name         varchar(64) check (name ~* '^[abcdefghijklmnopqrstuvwxyz .]+$'),
   /* A valid age is between 18 and 65 */
   age          int check (age > 17 and age < 66)
);
insert into fct.worker(name,age) values ('Joe F. Monkey', 18);
insert into fct.worker(name,age) values ('Sally G. Wonder', 65);

/* Question: Will this statement work?  Why or why not? */
insert into fct.worker(name) values ('Zebra has no age');

/*
 * Machine.  Note: We move the attributes in the weak entity (location)
 * to the machine table. We assume a simple (unrealistic) representation
 * of location, namely that machines can be on a floor and in one of 9
 * spots on a floor.
 */

 /*
```

```
  * We create a domain for representing floors.
  */
 create domain fct.floor_domain integer
    check (value in (
        0,  /* Unassigned */
        1,  /* Ground floor */
        2,  /* Higher */
        3   /* Highest */
));

/*
 * On a floor a machine may be located in one of nine places
 *          1 | 2 | 3
 *          4 | 5 | 6
 *          7 | 8 | 9
 *          0 -- unassigned spot
 */
 create domain fct.spot_domain integer
    check (value between 0 and 9);

CREATE TABLE fct.machine (
   mid               SERIAL PRIMARY KEY,
   model             varchar(64),
   /* Location of the machine, represented by floor and spot on floor */
   loc_floor         fct.floor_domain DEFAULT 0,
   loc_spot          fct.spot_domain DEFAULT 0,
   /* Informal description of the location */
   loc_notes         text,
   /* Foriegn key to the factory */
   fid               int references fct.factory(fid) on delete no action
);

/*
 * We assume that the a factory ID of 1 exists. Why can we assume this? (This is sketchy but it
 should work
 */
insert into fct.machine(model, loc_notes, fid) values ('Model A', 'Location unknown', 1);
insert into fct.machine(model, loc_floor, loc_spot, loc_notes, fid) values ('Model A', 3, 4,
'Location: F-3,S-4', 1);

/*
 * PowerReading. (Note how we create a composite primary key.)
 */
CREATE TABLE fct.PowerReading (
   mid               int references fct.machine(mid) on delete no action,
   t                 timestamp,
   kWh               numeric(10,2),
   PRIMARY KEY(mid,t)
);

insert into fct.PowerReading(mid,t,kWh) values (1, clock_timestamp(),111.1);
insert into fct.PowerReading(mid,t,kWh) values (1, clock_timestamp(),222.2);
insert into fct.PowerReading(mid,t,kWh) values (1, clock_timestamp(),333.3);
```

```
/*
 * Session.  Note that Session is a superclass.  When workers use a machine,
 * inspect a machine, or repair a machine we record this fact in Session.  In
 * addition, if necessary, we record additional information in the tables
 * Inspection and Repair.
 */
CREATE TABLE fct.Session (
   sid               SERIAL PRIMARY KEY,
   wid               int references fct.Worker(wid) on delete no action,
   mid               int references fct.Machine(mid) on delete no action,
   log_in_time       timestamp,
   log_out_time      timestamp,
   notes             text
);

CREATE TABLE fct.Inspection (
   sid               int references fct.Session(sid) on delete no action,
   rating            int,
   next_inspection   date,
   notes             text,
   PRIMARY KEY(sid)
);

CREATE TABLE fct.Repair (
   sid               int references fct.Session(sid) on delete no action,
   cost              money,
   notes             text,
   PRIMARY KEY(sid)
);
```