Impact of Globalization and GeoPolitical Issues

- On technology?
- On society?
- On Software development?
  - On the SW design
  - User interface is easy
- On your business model?
Global Software Development

- “software work undertaken at geographically separated locations across national boundaries in a coordinated fashion involving real time (synchronous) and asynchronous interaction”
  - Requires communication
  - Coordination of groups
  - Control of groups

Why GSD?

- Solving local IT skills shortage
  - Motorola 2000 had 20% of required staff for 3G trial
  - Alternate: remain focused on core competencies
- Cost saving
- New markets of presence in local market
- Competitive advantage
- “follow the sun” development process
- Improved quality ????
Extent of GSD

- Over 50% of Fortune 500 companies use GSD
- Upwards of 50 nations
- IBM, British Airways, Alcatel, British Telecom, GE, etc.
- 80% of Irish software industry’s output is exported
- Open source projects hosted at SourceForge

Challenges in GSD

- Strategic issues
  - Determination of how to “split” project
- Cultural issues
- Communication
- Knowledge management
- Project and process management
- Technical issues
  - Due to bandwidth problems, replicating code
Legal and Political

- Laws differ across nations
  - IP laws?
- Who do you (the company) negotiate with for licenses
  - Depends on the country: private, government
- Impact of political stability
  - Changes in government and cancelled licenses
- “tuning” your product to meet political restrictions
  - Google, Blackberry IM, etc.
- Cyberattacks – is this a new “cold war”
  - ATP attacks sponsored by governments?

Impact on Business Model

- New markets also introduce new challenges and constraints
- Cost constraints
  - iPhone vs Android
    - Status symbol vs utility
- Understanding market potential
  - Reach/market penetration can change costs
    - BBM service rates, CISCO strategy
- Design for customer base – becomes difficult
Security and Privacy of Data

Database Security
Databases provide

- Shared access
- Controlled access
- Data consistency
- Data integrity
- Minimal redundancy

DB Security Requirements

- **Secrecy**: Users should not be able to see things they are not supposed to.
  - E.g., A student can’t see other students’ grades.
- **Integrity**: Users should not be able to modify things they are not supposed to.
  - E.g., Only instructors can assign grades.
- **Availability**: authorized users should be able to see and modify things they are allowed to
DB Security

- Non-repudiation:
  - DB cannot deny getting requests for changes
- Auditability
  - Helps determine if inappropriate disclosure has occurred
  - Helps track divulged information to prevent inference
  - Difficult to record access of fields; often a field is reported to have been accessed when it has not – e.g. SELECT all entries with ZIP 20007

DB Security

- To achieve the objectives, a security policy should be developed
  - Determine data to be protected, user access to data
- Security mechanisms to enforce the policy
  - Note: when DB is a backend of an application, also need to be able to authenticate users
    - Logging in to Amazon through a network
    - Assure users they are communicating with amazon and not a malicious server masquerading as amazon
      - Have you seen this?
Means of Achieving Security Requirements

- Access Control
  - Inference is a problem (fields are related; knowledge of race or gender can be a good predictor of salary, for example)
  - Size and granularity different from access control in OS

- User Authentication
  - DBMS does its own authentication because no trusted path between DBMS and OS when DBMS is an application program on top of OS

Access Controls

- A security policy specifies who is authorized to do what.
- A security mechanism allows us to enforce a chosen security policy.
- Two main mechanisms at the DBMS level:
  - Discretionary access control
  - Mandatory access control
Discretionary Access Control

- Based on the concept of access rights or privileges for objects and mechanisms for giving users privileges (and revoking privileges).
  - In Databases, objects refer to Tables and Views
  - Creator of a table or a view automatically gets all privileges on it.
    - DMBS keeps track of who subsequently gains and loses privileges, and ensures that only requests from users who have the necessary privileges (at the time the request is issued) are allowed.

SQL GRANT Command

- The following privileges can be specified:
  - SELECT: Can read all columns (including those added later via ALTER TABLE command).
  - INSERT(col-name): Can insert tuples with non-null or non-default values in this column.
  - DELETE: Can delete tuples.
  - REFERENCES (col-name): Can define foreign keys (in other tables) that refer to this column.
- If a user has a privilege with the GRANT OPTION, can pass privilege on to other users (with or without passing on the GRANT OPTION).
- Only owner can execute CREATE, ALTER, and DROP.
GRANT and REVOKE of Privileges

- **GRANT INSERT, SELECT ON Sailors TO Trevor**
  - Trevor can query Sailors or insert tuples into it.
- **GRANT DELETE ON Sailors TO Kunal WITH GRANT OPTION**
  - Kunal can delete tuples, and also authorize others to do so.
- **GRANT UPDATE (rating) ON Sailors TO Sarah**
  - Sarah can update (only) the rating field of Sailors tuples.
- **GRANT SELECT ON ActiveSailors TO Guppy, Yuppy**
  - This does NOT allow the ‘uppies to query Sailors directly!
    - ActiveSailors is a View over Sailors and Reserves tables
- **REVOKE:** When a privilege is revoked from X, it is also revoked from all users who got it solely from X.

GRANT/REVOKE on Views

- If the creator of a view loses the SELECT privilege on an underlying table, the view is dropped!
- If the creator of a view loses a privilege held with the grant option on an underlying table, (s)he loses the privilege on the view as well; so do users who were granted that privilege on the view!
Views and Security

- Views can be used to present necessary information (or a summary), while hiding details in underlying relation(s).
  - Given ActiveSailors, but not Sailors or Reserves, we can find sailors who have a reservation, but not the bid’s of boats that have been reserved.
  - Creator of view has a privilege on the view if (s)he has the privilege on all underlying tables.
  - Together with GRANT/REVOKE commands, views are a very powerful access control tool.
- How about GUI based views
  - Take the user to a different GUI based on their role?..banner?

Role-Based Authorization

- In SQL-92, privileges are actually assigned to authorization ids, which can denote a single user or a group of users.
  - CREATE ROLE and DROP ROLE
  - GRANT role and REVOKE role
- In SQL:1999 (and in many current systems), privileges are assigned to roles.
  - Roles can then be granted to users and to other roles.
  - Reflects how real organizations work.
  - Illustrates how standards often catch up with “de facto” standards embodied in popular systems.
Security to the Level of a Field!

- Can create a view that only returns one field of one tuple. (How?)
  - Then grant access to that view accordingly.
- Allows for arbitrary granularity of control, but:
  - Clumsy to specify
  - Performance is unacceptable (Too many view creations and look-ups.)

Problem?

- Discretionary control has some flaws, e.g., the Trojan horse problem:
  - Jack creates Horsie and gives INSERT privileges to Peter (who doesn’t know about this).
  - Jack modifies the code of an application program, to update grades, used by Peter, to additionally write some secret data (Jack’s grades) to table Horsie.
  - Now, Jack can see the secret info.
- The modification of the code is beyond the DBMSs control, but it can try and prevent the use of the database as a channel for secret information.
Mandatory Access Control

- Based on system-wide policies that cannot be changed by individual users.
  - Each DB object is assigned a security class.
  - Each subject (user or user program) is assigned a clearance for a security class.
  - Rules based on security classes and clearances govern who can read/write which objects.
- Most commercial systems do not support mandatory access control. Versions of some DBMSs do support it; used for specialized (e.g., military) applications.

Intuition

- Idea is to ensure that information can never flow from a higher to a lower security level.
- E.g., If Narahari has security class C, Wood has class S, and the secret table has class S:
  - Narahari’s table, Horsie, has Narahari’s clearance, C.
  - Wood’s application has his clearance, S.
  - So, the program cannot write into table Horsie.
- The mandatory access control rules are applied in addition to any discretionary controls that are in effect.
Internet-Oriented Security

- **Key Issues:** User authentication and trust.
  - When DB must be accessed from a secure location, password-based schemes are usually adequate.
  - For access over an external network, trust is hard to achieve.
    - If someone with Kunal’s credit card wants to buy from you, how can you be sure it is not someone who stole his card?
    - How can Kunal be sure that the screen for entering his credit card information is indeed yours, and not some rogue site spoofing you (to steal such information)? How can he be sure that sensitive information is not “sniffed” while it is being sent over the network to you?
- *Encryption* is a technique used to address these issues.

Encryption

- Encryption also provides privacy
- Plain text is *encrypted* using a *key* into *cipher text*
  - Only the user with the correct key can decrypt the cipher text
- Does encryption solve all security and privacy problems?