Managing Development of University Information Infrastructure: the Challenges and the Limits

Jana Kohoutková

Institute of Computer Science, Masaryk University in Brno, Czech Republic
kohoutkova@ics.muni.cz

Abstract

A typical information infrastructure of nowadays University consists of several independently existing databases and, built on top of them, application systems. To keep the development of such structure within the limits of a cross-linked and consistent whole needs to adopt and apply various management, organizational and technological principles and rules. The topic of this paper is to discuss the possibilities and limits of applying various dimensions of systems integration to university information infrastructure development. The discussion ranges from management and organization principles to the ICT ones and takes into account typical structures of university information systems users and providers.

Keywords: University information infrastructure, systems integration taxonomy.

1 Introduction

University information infrastructure (*UII*) generally consists of a whole series of information systems (*IS*) processing specialised data areas of both university-wide and local extents. Systems supporting the areas of:

- *study & teaching*,
- *science & research*,
- *economics & accounting*,
- *human resources* (i.e., *personnel & wages*),
- *public relations*,
- *legal services*, and
- *technical & operational services*

are identified as the key ones for university life. The clear need is to cross-link these key systems into a consistent, integrated management and information system offering quality information services to highly structured user community. The integrated system is built in co-operation of internal teams of (so called) *methodical* professionals and *ICT* professionals, as well as *external* IS providers. The possible extent and dimensions of this consistency are discussed in this paper drawing from experience gathered from building an integrated UII at Masaryk University in Brno.

2 Management and Organizational Principles for UII Development

As stated in the previous section, UII consists of:

- *university-wide databases and application systems*, forming the *UII core* and providing information support in the few identified university-specific areas. These systems need to be under central management;
- *local databases and application systems*, serving only to selected groups of departments. These systems usually remain outside central management.

Central management of UII development has to consider a) *users* and *providers*, b) *general management principles*, and c) *collaboration principles for external IS providers*.

2.1 UII Users and Providers

The conception, analysis, development, operation and integration of individual IS forming the UII have two sides:

- the *user* (*customer*) *side* responsible for the collection, analysis and specification of user requirements, and also for testing and approving the functionality of final systems;
- the *provider side* responsible for a) user requirements analysis (in collaboration with the user side), b) overall system architecture analysis and design, and c) the analysis, design, implementation and integration of individual systems.

At the university level, i.e. in the scope of university-wide databases and applications, the UII *user side* is grouped into the following basic levels according to the extent of access rights to data and applications:

- *R … Rector's Office*, i.e., head managers and specialised users at university-wide level (specialised users being from specialised departments – for studies, R&D, economy, personnel, etc.) whose access rights to internal (non-public) information cover the whole university,
- *S … Schools and Faculties*, i.e. head managers and specialised users at the level of individual schools and faculties whose access rights to internal information cover the respective school/faculty,

- *D … Departments*, i.e. head managers, secretariats and specialised users at the departmental level with access rights to internal information limited to the respective department,
- *P … Persons*, i.e. individual employees, students a external collaborators whose access rights to internal information are limited to their own personal data,
- *W … World*, i.e. the public having no access to internal information.

The *provider side* is represented by:

- *external organizations/companies*;
- *internal ICT teams*.

2.2 General Management Principles

A necessary condition of a successful UII development is clearly stated and distributed responsibilities both on the user side (co-ordination of user requirements) and the provider side (co-ordination of realization) as well as mutually between them. User side co-ordination must be ensured at two levels:

- *internal* = communication with users in individual areas of specialization (study, R&D, …), both *in-depth* (across specialized departments in a given area to individual end-users) and *in-width* (in connection to other related specialised areas);
- *external* = communication with the provider side co-ordinator.

Provider side co-ordination must be ensured at similar two levels:

- *internal* = communication with providers, i.e. co-ordination of various layers of systems integration (will be discussed in Sections 3 and 4);
- *external* = communication with the user side co-ordinator.

User side co-ordinators for individual areas of specialization are called *methodists*, provider side co-ordinators are called *systems administrators* or *ICT-coordinators*. Because of the integration needs, the *general co-ordinator* should be the head *ICT-coordinator*.

2.3 Collaboration Principles for External Providers

Usually it is not possible to develop the UII utilizing exclusively internal capacities. Rather, some external providers must be hired. The dividing line between "in-house" and "external" UII development is never crisp. A reasonable criterion is to hire external bodies to deliver those UII parts that are well within their subjects of specialisation (typically: economic information systems or systems for human resources), and to save internal capacities to:

- building applications that are university-specific (*intensive level development*);
- building applications complementing the functionality of externally provided specialized applications by delivering information processed in the external systems to other end-users (potentially to the whole academic community) via university intranet systems (*extensive level development*).

The users of the external systems should be from levels *R* and *S* respectively, while user levels *P* and *W* are typical customers for in-house developed systems (needless to say, university intranet as well as university internet presentation are typical candidates for in-house development, see further Subsection 4.2). The user level *D* is on the boundary, more inclining to be a customer to in-house developed systems.

Collaboration with external providers must be driven by a few principles directly derived from Subsection 2.2. The following roles must be appointed:

- *general co-ordinator* on the *university* side (head ICT co-ordinator, usually named in the contract);
- *general co-ordinator* on the *external provider* side (usually named in the contract);
- *contact persons* for partial areas of collaboration on the *university* side. They are a) the *methodists* for the respective application areas and b) the *systems administrators* for the ICT areas;
- *contact persons* (so called *consultants*) for partial areas of collaboration on the *external provider* side.

On the external provider side the collaboration rests in *providing* regular/standard *support services* (hot-line etc.) and *claims solving* (fixed part of the collaboration), and in *providing* special *services on demand* (variable part of the collaboration). Standard support in methodical or ICT areas is provided via the respective contact people (the methodists guarantee that consultations and training are further propagated to end-users), and similarly claims solving of incorrect functioning of the externally provided system. Services on demand (usually phased to *Requirement – Order – Realization – Supply – Acceptance – Payment*) are also co-ordinated by the respective contact people but in this case *approval* of the university general co-ordinator is required to ensure the UII development integrity.

3 Systems Integration Taxonomy

In order to describe the different ICT dimensions of information systems integration, three basic aspects are usually identified in accordance with the classification reported in [4] – *data*, *control*, and *presentation* integration. The classification discussed in [5] identifies two additional aspects – *platform* and *process* integration, and some other classifications add also *communication* integration.

Data integration deals with the management of data design in a consistent (cooperative) way. *Control integration* (alias *functional* or *application integration*) deals with the automatic coordination of system activities (system functionality). The subject of *presentation integration* is to support a homogeneous user interface to the information provided by the system (more generally: to all services of the system). *Platform integration* represents services for network and operating system transparency, and *process integration* refers to the support of the system development process. *Communication integration* supports teamwork within and around system development.

Of the wide range of ICT requirements laid on modern information systems there are two that are the basic ones from the users' viewpoint. They are:

- quality *information contents*, and
- quality *access* to information.

Viewing the quality of university information infrastructure from the users' point we may (and shall in this text) stay less concerned with platform transparency, system development process, or system development communication. Instead, we shall concentrate on *data contents* (i.e., *data integration*), and *data access* (i.e., *application* and *presentation integration*, supplemented by *authentication & authorization integration*).

4 Technological Principles for UII Development

4.1 Data Integration

The *UII data level* is formed by university-wide databases and local databases the former being subject to central administration. The task of central database administrators is to conceive and implement the structure and mutual connections of university-wide databases. An obvious part of the task is the choice of suitable data management technologies (relational databases, XML data file systems, …) together with the supporting tools (e.g., relational database machines).

Typically, various central data sources (generally autonomous and distributed) overlap in data contents (a frequent problem is multiple stores of personal data). There are various well-known ICT methods and approaches to maintain data consistency in these cases. The following basic principles are identified in [1]:

- *specification of reference vs. mirror data sources* for the overlapping data areas;
- *definition of data standards*;
- *formal description of data models and transformations*; and
- *definition of access rights*, i.e., the *rules* governing the acceptance or rejection of data modifications in the reference data sources.

If a *reference database* is integrated with a *mirror database* uni-directional replication is used building on the *master-slave* principle[1]. If two *reference databases* are integrated bidirectional replication is used building on the *peer-to-peer* principle[2] (with manual clean-up of data inconsistencies if the automated integration process fails).

Data for central university systems is hardly ever stored in one database only. It is therefore necessary to count with multiple occurrence of the same information and with the necessity to maintain these multiple sources in consistent state. Because of high demands of integration processes implementation and maintenance it is recommended to pursue *master-slave* integration (in systematic, regular regime, utilizing data standards wherever applicable) and to strictly avoid *peer-to-peer* integration unless it is really inevitable.

A few recommendations for managing development of the UII data layer are:

1. *limited scale of technologies used:* The scale of database technologies should be limited – to cover the requirements while allowing necessary specialisation (detail knowledge) of the administrators;
2. *limited number of data stores:* The number of central data stores should be limited – to provide the required performance and to separate data according to their privacy level (internal-private versus external-public data);
3. *central management of data stores:* The distribution of central databases to individual data servers requires central management. Ideally, data of a certain area should be stored in one place only, and from there provided to various applications. If for some reason (applications response time, data store security, etc.) the data needs to be replicated then the master-slave principle is strongly preferred (see above). The peer-to-peer replication should be minimised to the least possible extent. The integration of replicated data needs to be strictly demanded by the central management;
4. *general interfaces for data provision:* Data stored in one place and used by multiple application systems should be accessible via suitable – open and fully documented – APIs replacing direct reads and writes from/to the respective data store;

[1] Master-slave principle means that all data modifications are primarily done in reference database and from there passed to mirror database. The principle must be strictly respected even if the modifications are initialised by applications running over the mirror database – whether the modification is allowed or not is always the decision of the reference database. If the modifications are initiated by mirror database they must be recorded in both databases on-line. Other modifications made in the reference database may be recorded in the mirror database with some delay.

[2] Peer-to-peer integration is driven by centrally defined rules.

5. *robust solution of data stores:* The central data store should be protected from possible hardware failures, at least by the existence of a back-up hardware ready to be switched to within a short time in case the primary data store hardware fails, ideally by cluster configuration of the hardware.

4.2 Application and Presentation Integration

Similarly to data integration, the *UII application and presentation level* consists of university-wide applications and application systems (providing access to data in central databases) and local applications. The task of central application administrators is to conceive and implement the structure and mutual connections (i.e., application program interfaces – APIs) of university-wide applications. An obvious part of the task is the choice of suitable application development technologies (Java, .NET, …) together with the supporting tools (e.g., application servers and application development tools).

Data access quality is measured by "any time from anywhere timely" as well as "fast and efficient work", and these need to be compromised depending on the type of the user. Two types of access should be combined:

- access via *web clients,* and
- access via *specialized non-web clients*.

Non-web access is always *internal*, devoted to limited number of authorised users. It is usually available from selected workstations only, and provides special, more demanding data-processing functions and operations (write-operations and complex read-operations) that are not needed, or even not wished, to be available in a larger extent.

The clear advantage of web access is the identical requirement on client software (i.e., a common web browser) allowing access from any Internet-connected workstation at any time irrespectively of how much the underlying database and application technologies differ. Web access is both *internal* (university intranets) and *external* (university public presentation). *Web intranet systems* provide secured and differentiated access to data to (as many as possible) authorised people to whom the data refers or who need the data for their work – ideally to all active academic community members. The access is both for read and write, the latter in simpler operations only. *Web internet systems* can be viewed as generalised intranets in the sense that the users concerned are the public in general, hence no security and differentiation arrangements are needed. The access is mainly for read (only exceptionally for write) arranged in simple and fast-processed operations.

In Subsection 2.1 the basic user levels have been identified. At all those levels the users need read-access to data, in the extent and with security arrangements respecting the measure of information privacy. The question is how to best combine the above described two types of user access with the identified user levels. A compromise solution, already formulated as a result of discussion in Subsection 2.3, is:

R … non-web internal access,

S … non-web internal access,

D … web internal access,

P … web internal access,

W … web external access.

From this it follows that the task of web intranet systems is to: a) complement the functionality of specialized non-web application systems in the "*extensive*" sense (by delivering information processed by non-web applications to other end-users who do not fall into the category of non-web systems authorised users); b) complement the functionality of specialized non-web application systems in the "*intensive*" sense (by providing other information services that are university-specific and/or aimed at user levels *D* and *P* respectively); c) provide automatic data maintenance services (via various control mechanisms running over central data stores).

In cases that the functionalities of various non-web and web systems overlap (similarly to overlapping data stores discussed in Subsection 4.1) there are various ICT methods and approaches available to maintain application consistency (web services etc.). If external providers are employed the basic requirement laid on the delivery of their systems should be "*open and fully documented APIs*" providing necessary commands, functions, protocols and tools for building cross-linked software applications without the need to know internal data and application structures.

In cases that the functionalities of various web systems overlap the question is how to optimise the presentation level, i.e. when to just use hyperlinks to another system, and when to duplicate the same information in different user interfaces (a typical example being personal contact information). The following principles are recommendable:

- ideally there should be only one university-wide intranet serving ordinary end-users to access (and maintain) internal information. If the ideal is not available then "the fewer the better" holds;
- internal and external web systems should differ sufficiently in their presentation layers to make it clear to the users whether the provided information is of private or public nature, or is a private or a public version of the same information;
- a piece of information that has a principal importance in a particular web system should be implemented in the presentation logic of this system (i.e., in a unified browsing=navigational logic and viewing=graphical logic) rather then using hyperlinks to another system providing the same information in different browsing & viewing logic. On the contrary, information of a marginal importance already existing in another web system should be provided via a hyperlink to that other web system. The difference between principality and marginality is rather fuzzy. Some hints may be driven from the number of links to the shared piece of

information from a given web system, or the probability that the user who needs the referred information will immediately return to the referring place – the higher is the number or probability the more principal the information is.

In summary, the recommendations for managing development of the UII application and presentation layer are:

1. *limited scale of technologies used:* The scale of technologies used for application development should be limited – to cover the requirements while allowing necessary specialisation (detail knowledge) of the developers' teams;
2. *web vs. non-web client:* Head managers and specialised users at both university-wide and faculty levels should be equipped with *non-web clients* (thick clients, thin Java clients) to access information – because of the response time and efficiency. The academic community as a whole (with differentiated access rules according to individual responsibilities) should be equipped with university *intranet with web clients*. The world (unlimited access rights) should be equipped with university *internet presentation*, i.e., *web clients* again;
3. *internal vs. external provider:* Information services requiring specialised knowledge of a given application area and having good support at the software market are candidates for external provision (typically non-web solutions). In contrary, information services that are specific and require adaptation to special requirements of the university are candidates for in-house development (typically web solutions);
4. *limited number of application systems:* The number of university-wide application systems should be if possible limited – while preserving necessary independence of distributed developers' teams (internal, external). Specialised applications serving specialised users for their daily work may differ from each other and it is not necessary to struggle for their unification. On the other hand, university intranet system addressed to the whole academic community should be only one, as well as the internet presentation (the latter allowing individual faculties/departments/people to incorporate their local presentation pages);
5. *information support completeness:* University-wide databases and application systems should exist for all basic areas of university activities (see Section 1);
6. *general interfaces for functionality provision:* A functionality implemented in one application and utilised in other applications (in the form of pre-processed data outputs) should be accessible via suitable – open and documented – interface. This means that efficient collaboration must exist between various development teams (both external and internal ones), and agreements on data provision interfaces – both for atomic data stored in databases, and data pre-processed by applications;
7. *differentiating vs. unifying presentation interfaces:* Web subsystems using different navigation logic should be distinguished by different graphics;
8. *central orientation – web application map:* Multiple web subsystems (both internet and intranet ones) should be supplemented by a centrally maintained joint index (map) of applications.

4.3 Authentication and Authorization Integration

For *authentication* it is essential, and quite often the way the UII works, if all intranets use one common database of user entry logins and passwords built on top of central personnel database. To increase security there exist various methods of authentication provision in information systems, see, for example, discussion in [2].

From the *authorization* viewpoint the users are divided into groups according to access rights to data and applications. Access rights are basically of two types:

- *implicit* ones, following from the relationship of a person to the university, typically *student*, *department head*, *head academician*, *head of a specializes department*, etc. An implicit access right is usually related to certain department at some hierarchical level (*dean of the faculty*, *head faculty economist*, *employee of a department*, *student at a faculty*, etc.);
- *explicit* ones, specifically delegated for various application-specific – and often time limited – activities (*project leader* etc.). An explicit access right may or may not be related to a department.

Similarly to authentication, it is essential if the authorisation is maintained centrally. This is not a problem for implicit access rights that are usually derived from central personnel database, but explicit rights are often application-system-specific and integration mechanism based on APIs must be employed to keep them UII-consistent (see also discussion in Subsection 4.2). It is recommended that lists of explicit user groups (explicit roles) are made available in the intranet systems, and the validity of explicit access rights is driven and automatically controlled by suitable systems of rules[3].

5 Closing Remarks

UII is a very live organism requiring for its development and maintenance permanent quality work and collaboration of various specialised groups of people – under quality coordination.

[3] Security aspects are outside the scope of this paper nonetheless it is worth mentioning that various application-driven security arrangements should be implemented in the intranet systems such as so called IP-security discussed in [3].

The paper draws from the experience in building UII at Masaryk University in Brno combining both in-house built and externally provided information subsystems. Principles and approaches discussed here have been applied and proved correct and worth further extending and generalising. Currently they are imposed on potential external providers in a tender for a new Information System for Human Resources within the integrated UII.

References

[1] J. Kohoutková. "University Information Infrastructure – An Integration Challenge", *Proc. of EUNIS 2002*, Porto (PT), pp. 268–271, (2002).

[2] J. Ocelka. "Authentication Provision in Information Systems", *Proc. of DATAKON 2003*, Brno (CZ), pp. 161–169, (2003).

[3] J. Ocelka, J. Měcháček. "User Security of Information Systems", *Proc. of Tvorba softwaru 2002*, Ostrava (CZ), pp. 160–164, (2002).

[4] D. Schefström, G. van den Broek (eds.). *Tool Integration: Environments and Frameworks*. John Wiley & Sons, pp. 434, (1993).

[5] A. I. Wasserman. "Tool Integration in Software Engineering Environments", *Proc. of the International Workshop on Environments*, LNCS 467, Springer-Verlag, pp. 137–149, (1990).