1. Below is an article in the Washington Post, which describes the situation where the Internet and information technologies may benefit from the recent rapid increase of oil price. Please read the article carefully and answer the following two questions in English:

(1) Please write a 600-word summary (in your own words) to describe the major points mentioned in the article and how different kinds of websites can benefit from the increase of oil process. Note: the original article has about 4500 words. In other words, the summary you write is roughly 1/7 in length of the original article. (20%)

(2) Please identify two potential research issues in information systems (may be technical or managerial) from the description of the article. For each research issue, briefly describe why you think the issue is important and how you would conduct the research. (30%)

Note: Your answer must be written in English. No score will be given if your answer is written in Chinese.

Pumped-Up Comparison Shopping
Search for Cheap Gas Increases Web Traffic, Inspires New Services

By Yuki Noguchi, Washington Post Staff Writer

A new service from Verizon Wireless lets customers use cellphones to search for the cheapest, closest gas stations. To get FuelFinder, subscribers have to pay $1.99 a month, in addition to a $5 monthly Web access fee.

But what's a few bucks, if you can save a few cents?

"It saves people from having to drive for miles to look for cheap gas," Verizon Wireless spokesman John Johnson said.

Sharing information on how to deal with gasoline prices has driven huge amounts of traffic on Web sites in recent weeks, as drivers fume, exchange tips on where to buy gas and increase fuel efficiency and even gather online to pray for lower gas prices.

Last month, gas-price-tracking Web sites posted some of highest growth in traffic on the Internet, according to ComScore Media Metrix. The most popular, Gasbuddy.com, compiles user-reported prices at stations around the country, and got 2.3 million visitors in April, up from 755,000 in March. The Energy Department's site, Fueleconomy.com, saw a 172 percent increase the same month.

Gasbuddy.com now gets from 1 million to 1.5 million visitors a day, and hosts forums on which 16 million messages have been posted, said Jason Toews, a co-founder of the organization, based in Brooklyn Park, Minn..

There, in addition to kvetching about cost, people hold forth about alternative fuels, suggest taking heavy loads out of the car to save gas and advocate driving slower on the highway to get better gas mileage.

"I've noticed people are driving slower," said Toews, who said it's possible to increase the efficiency of a car by 20 percent by going 55 mph instead of 75.

Other sites -- such as GasPriceWatch.com, Gaswatch, GasPriceAlert, and the Utility Consumers' Action Network's Gasoline Price Tracking Service -- also help consumers comparison shop, sometimes by compiling data from customers themselves who report to the site. FuelFinder compiles data reported by gas stations and credit card companies and the service allows subscribers to choose between unleaded, premium mid-grade or diesel prices.

Search engines also provide an increasingly popular way to find fuel-related information. Internet searches for the term "ethanol" increased 212 percent from April to May, according to another research company, Hitwise. Searches for "biodiesel" increased 100 percent during the same period.

During the past month, Yahoo Inc. said the number searches on its site for gas prices and hybrids increased 250 percent, peaking in the last week of April, coinciding with the peaking of gas prices and Earth Day on April 22.

On the Internet overall, Toyota Motor Corp. and Honda Motor Co. -- two Japanese automakers that make popular models of hybrid cars -- saw a boost in Web site visitors in April, according to ComScore, with traffic on those pages up 49 percent and 33 percent, respectively, over the previous month.

Bloggers are getting a lot of mileage out of the topic, as well, using social networking techniques to seek out cheaper gas.

"I just [paid] $47 to fill up my little Hyundai. This is nuts," a MySpace.com blogger in Massachusetts, who called himself Troy aka Cockroach, wrote earlier this week. "So, I managed to find it for $2.99 . . . up at the Hess. Anyone know any cheaper spots around [here]? I heard S. Main st. Hess is even cheaper."

Some are aiming higher than fellow Web surfers for help. On Wednesday, online worship center Praylive.com hosted a 3,295-participant group prayer for lower gas prices.

"It affects all of us, we all need gas," said Wenda Royster, founder of Baltimore-based Pray Live, who said one woman she spoke to is selling her own blood to buy gas. Royster wants to bring people together to carpool and to affirm the power of prayers, she said. "When we prayed in DC [on April 27], gas prices came down four or five cents."

Renee, a MySpace blogger from Delaware, Ohio, offered up MSN's gas-tracking Web site address on her site. "Be a good neighbor and pass this along," she wrote.

Verizon hopes to cut through such informal efforts with FuelFinder, the cellphone service, which allows users to type in a Zip code or town name to pull up a map and a short list of nearby stations, starting with the cheapest. The site comes with links for turn-by-turn directions to each station.

FuelFinder's database has 1,397 stations in the Washington area. A search yesterday for the cheapest location in downtown Washington showed a Hess Station on 1739 New Jersey Ave. NW selling at a reported $3.06 per gallon of unleaded gas. According to FuelFinder, that is 16 cents more than the national average of $2.90, but 15 cents less than the Amoco Station at 45 Florida Ave. NE, about two-thirds of a mile away.

FuelFinder's overall database is compiled by MobileGates Corp., and includes prices from more than 110,000 service stations, but does not include discount shopping clubs, such as Costco or Sam's Club.

In rare cases, savings on gas can total 30 cents to 40 cents per gallon in some cities, but the bulk of stations run at around the same price, which means you can win the hunt for cheap gas, but lose money overall by paying for finder services or driving out of your way, said consumer-behavior expert Terry Childers, a professor at the University of Kentucky. "But sometimes the emphasis is on rewards, and you forget about the cost."

國立中山大學95學年度博士班招生考試試題

科目：資訊科技論文評述(一)【資管系選考】

共 頁第 頁

Read the attached paper carefully and answer the following questions.

1. The authors proposed another ERD as shown in Figure 3a to replace that shown in Figure 2 so as to avoid using new constructs. Do you think these two ERDs are equivalent? If not, show a set of instances that are valid in one ERD but not valid in another. (10%)
2. Suppose we want to enforce a constraint that a staff person can only work in a branch operated by her/his division. Can this constraint be captured by the ERD shown in Figure 6? Please justify. (10%)
3. Show a set of example cardinality constraints that cannot be captured using the current ER Model. (10%)
4. Do you agree that relationship constraints and attribute constraints in general cannot be captured using ERD? Please show a new example. (15%)
5. Please criticize this paper. (5%)

Entity-Relationship Modeling *Re*-revisited

Don Goelman[1] and Il-Yeol Song[2]

[1] Department of Computer Science
Villanova University
Villanova, PA 19085
don.goelman@villanova.edu

[2] College of Information Science and Technology
Drexel University
Philadelphia, PA 19104
song@drexel.edu

Abstract. Since its introduction, the Entity-Relationship (ER) model has been the vehicle of choice in communicating the structure of a database schema in an implementation-independent fashion. Part of its popularity has no doubt been due to the clarity and simplicity of the associated pictorial Entity-Relationship Diagrams ("ERD's") and to the dependable mapping it affords to a relational database schema. Although the model has been extended in different ways over the years, its basic properties have been remarkably stable. Even though the ER model has been seen as pretty well "settled," some recent papers, notably [4] and [2 (from whose paper our title is derived)], have enumerated what their authors consider serious shortcomings of the ER model. They illustrate these by some interesting examples. We believe, however, that those examples are themselves questionable. In fact, while not claiming that the ER model is perfect, we do believe that the overhauls hinted at are probably not necessary and possibly counterproductive.

1 Introduction

Since its inception [5], the Entity-Relationship (ER) model has been the primary approach for presenting and communicating a database schema at the "conceptual" level (i.e., independent of its subsequent implementation), especially by means of the associated Entity-Relationship Diagram (ERD). There's also a fairly standard method for converting it to a relational database schema. In fact, if the ER model is in some sense "correct," then the associated relational database schema should be in pretty good normal form [15]. Of course, there have been some suggested extensions to Chen's original ideas (e.g., specialization and aggregation as in [10, 19]), some different approaches for capturing information in the ERD, and some variations on the mapping to the relational model, but the degree of variability has been relatively minor. One reason for the remarkable robustness and popularity of the approach is no doubt the wide appreciation for the simplicity of the diagram. Consequently, the desirability of incorporating additional features in the ERD must be weighed against the danger of overloading it with so much information that it loses its visual power in communicating the structure of a database. In fact, the model's versatility is also evident in its relatively straightforward mappability to the newer Object Data Model [7]. Now admittedly an industrial strength ERD reflecting an actual enterprise would necessarily be some order of magnitude more complex than even the production numbers in standard texts [e.g., 10]. However, this does not weaken the ability of a simple ERD to

P. Atzeni et al. (Eds.): ER 2004, LNCS 3288, pp. 43–54, 2004.

capture local pieces of the enterprise, nor does it lessen the importance of ER-type thinking in communicating a conceptual model.

Quite recently, however, both Camps and Badia have demonstrated [4, and 2 (from whose paper the title of this one is derived)] some apparent shortcomings in the ER model, both in the model itself and in the processes of conversion to the relational model and its subsequent normalization. They have illustrated these problems through some interesting examples. They also make some recommendations for improvements, based on these examples. However, while not claiming that the ER model can be all things to all users, we believe that the problems presented in the examples described in those two papers are due less to the model and more to its incorrect application.

Extending the ERD to represent complex multi-relation constraints or constraints at the attribute level are interesting research topics, but are not always desirable. We claim that representing them would clutter the ERD as a conceptual model at the enterprise level; complex constraints would be better specified in a textual or language-oriented format than at the ERD level.

The purpose of this paper is to take these examples as a starting point to discuss the possible shortcomings of the ER model and the necessity, or lack thereof, for modifying it in order to address them. We therefore begin by reviewing and analyzing those illustrations. Section 2 describes and critiques Camps' scenarios; Section 3 does Badia's. Section 4 considers some related issues, most notably a general design principle only minimally offered in the ER model. Section 5 concludes our paper.

2 The Camps Paper

In [4], the author begins by describing an apparently simple enterprise. It has a straightforward ERD that leads to an equally straightforward relational database schema. But Camps then escalates the situation in stages, to the point where the ER model is not currently able to accommodate the design, and where normalizing the associated relational database schema is also unsatisfying. Since we are primarily concerned with problems attributed to the ER model, we will concentrate here on that aspect of the paper. However, the normalization process at this point is closely tied to that model, so we will include some discussion of it as well. We now give a brief recapitulation, with commentary.

At first, Camps considers an enterprise with four ingredients: **Dealer**, **Product**, **State**, and **Concession**, where **Concession** is a ternary relationship among the other three, implemented as entity types. Each ingredient has attributes with fairly obvious semantics, paraphrased here: d-Id, d-Address; p-Id, p-Type; s-Id, s-Capital; and c-Date. The last attribute's semantics represents the date on which a given state awards a concession to a given dealer for a given product. As for functional dependencies, besides the usual ones, we are told that for a given state/product combination, there can only be one dealer. Thus, a minimal set of dependencies is as follows:

$$
\begin{array}{ll}
\{s\text{-}Id, p\text{-}Id\} \rightarrow d\text{-}Id & \\
\{s\text{-}Id, p\text{-}Id\} \rightarrow c\text{-}Date & \\
d\text{-}Id \rightarrow d\text{-}Address & \qquad (A)\\
p\text{-}Id \rightarrow p\text{-}Type & \\
s\text{-}Id \rightarrow s\text{-}Capital &
\end{array}
$$

2/12

An ERD for this is given in Figure 1 (attributes are eliminated in the figures, for the sake of clarity), and the obvious relational database schema is as follows:

State(s-Id, s-Capital
Product(p-Id, p-Type)
Dealer(d-Id, d-Address) (B)
Concession(s-Id, p-Id, d-Id, c-Date)

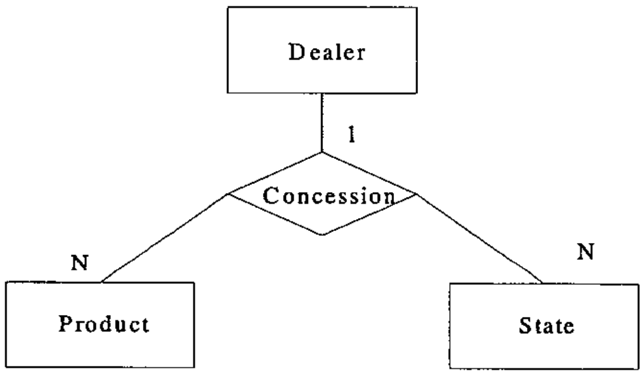

Fig. 1. Example of 1:N:N relationship (from Figure 1 in [4], modified)

The foreign key constraints derive here from the two components of Concession's key, which are primary keys of their native schemas. Since the only functional dependencies are those induced by keys, the schema is in BCNF. Here Camps imposes further constraints:

$p\text{-}Id \rightarrow d\text{-}Id$
$s\text{-}Id \rightarrow d\text{-}Id$

In other words, if a product is offered as a concession, then it can only be with a single dealer **regardless of the state**; and analogously on the state-dealer side. The author is understandably unhappy about the absence of a standard ERD approach to accommodate the resulting *binary constraining relationships* (using the language of [12]), which he renders in a rather UML-like fashion [17], similar to Figure 2. At this point, in order to highlight the generic structure, he introduces new notation (A, B, C, D for State, Dealer, Product, Concession, respectively). However, we will keep the current ones for the sake of comfort, while still pursuing the structure of his narrative. He notes that the resulting relational database schema includes the non-3NF relation schema Concession(s-Id,p-Id,d-Id,c-Date). Further, when Camps wishes to impose the constraints that a state (respectively product) instance can determine a dealer if and only if there has been a concession arranged with some product (respectively state), he expresses them with these conditions:

$$\pi_{s\text{-}Id,d\text{-}Id}(\text{Concessions}) = \pi_{s\text{-}Id,d\text{-}Id}(\text{State})$$
$$\pi_{p\text{-}Id,d\text{-}Id}(\text{Concessions}) = \pi_{p\text{-}Id,d\text{-}Id}(\text{Product}) \quad \text{(C)}$$

Each of these can be viewed as a double inclusion dependency and must be expressed using the **CHECK** construct in SQL.

3/12

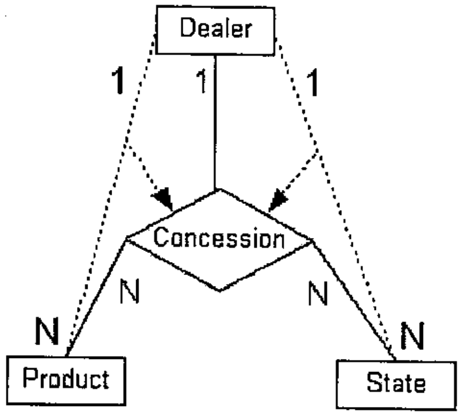

Fig. 2. Two imposed FDs (from Figure 2 of [4])

Now we note that it is actually possible to capture the structural properties of the enterprise at this stage by the simple (i.e., ternary-free) ERD of either Figure 3a [13] or Figure 3b [18]. The minimal set of associated functional dependencies in Figure 3a is as follows:

s-Id → s-Capital
p-Id → p-Type
d-Id → d-Address
s-Id → d-Id (D)
p-Id → d-Id
{s-Id, p-Id} → c-Date

One, therefore, obtains the following relational database schema, which is, of course, in BCNF, since all functional dependencies are due to keys:

State(<u>s-Id</u>,s-Capital,d-Id)
Product(<u>p-Id</u>,p-Type,d-Id)
Dealer(<u>d-Id</u>,d-Address) (E)
Concession(<u>s-Id,p-Id</u>,c-Date)

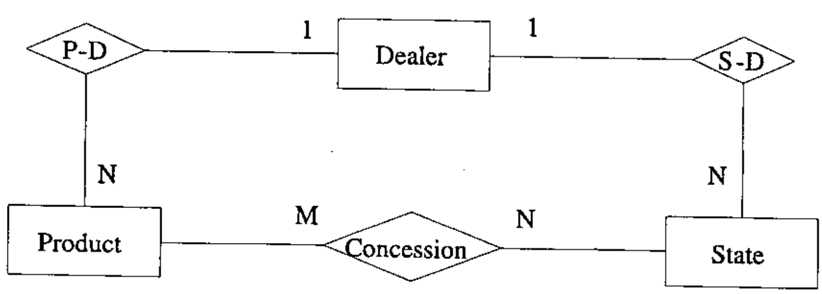

Fig. 3a. A binary model of Figure 2 with Concession as a M:N relationship

4/12

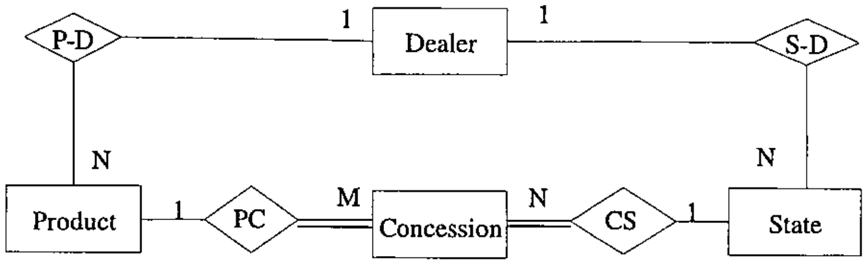

Fig. 3b. A binary model of Figure 2 with Concession as an intersection (associate) entity

Admittedly, this approach loses something: the ternary character of **Concession.** However, any dealer-relevant information to a concession instance can be discovered by a simple join; a view can also be conveniently defined. The ternary relationship in Figure 2 is therefore something of a red herring when constraining binary relationships are imposed to a ternary relationship. In other words, it is possible that an expansion of the standard ERD language to include n-ary relationships' being constrained by m-ary ones might be a very desirable feature, but its absence is not a surprising one.

Jones and Song showed that the ternary schema with FDs imposed in Figure 2 can have lossless decomposition, but cannot have an FD-preserving schema (Pattern 11 in [13]). Camps now arrives at the same schema (E) (by normalizing his non-3NF one, not by way of our ERD in Figure 3a). The problem he sees is incorporating the semantics of (C). The constraints he develops are:

$$\pi_{\text{s-Id, p-Id}}\text{ (Concessions)} \subseteq \pi_{\text{s-Id, p-Id}}\text{ (State*Product)}$$
$$\pi_{\text{s-Id}}\text{ (State)} \subseteq \pi_{\text{s-Id}}\text{ (Concessions) iff State.d-Id is not null} \qquad \text{(F)}$$
$$\pi_{\text{p-Id}}\text{ (Product)} \subseteq \pi_{\text{p-Id}}\text{ (Concessions) iff Product.d-Id is not null}$$

The last two conditions seem not to make sense syntactically. The intention is most likely the following (keeping the first condition and rephrasing the other two):

$$\pi_{\text{s-Id, p-Id}}\text{ (Concessions)} \subseteq \pi_{\text{s-Id, p-Id}}\text{ (State*Product)}$$
$$(\forall s_0 \in \pi_{\text{s-Id}}(\text{State}))(s_0 \in \pi_{\text{s-Id}}(\text{Concessions}) \text{ iff } (\exists d_0)(<s_0,d_0> \in \pi_{\text{s-Id, d-Id}}(\text{State}))) \qquad \text{(G)}$$
$$(\forall p_0 \in \pi_{\text{p-Id}}(\text{Product}))(p_0 \in \pi_{\text{p-Id}}(\text{Concessions}) \text{ iff } (\exists d_0)(<p_0,d_0> \in \pi_{\text{p-Id, d-Id}}(\text{Product})))$$

At any rate, Camps shows how SQL can accommodate these conditions too using **CHECKs** in the form of ASSERTIONS, but he considers any such effort (to need any conditions besides key dependencies and inclusion constraints) to be anomalous. We feel that this is not so surprising a situation after all. The complexity of real-world database design is so great that, on the contrary, it is quite common to encounter a situation where many integrity constraints are not expressible in terms of functional and inclusion dependencies alone. Instead, one must often use the type of constructions that Camps shows us or use triggers to implement complex real-world integrity constraints.

3 The Baida Paper

In his paper [2] in turn, Badia revisits the ER model because of the usefulness and importance of the ER model. He contends that, as database applications get more

5/12

complex and sophisticated and the need for capturing more semantics is growing, the ER model should be extended with more powerful constructs to express powerful semantics and variable constraints. He presents six scenarios that apparently illustrate some inadequacies of the ER model; he classifies the first five as **relationship constraints** that the model is not up to incorporating and the sixth as an **attribute constraint**. We feel that some of the examples he marshals, described below in 3.3 and 3.6, are questionable, leading us to ask whether they warrant extending the model. Badia does discuss the down side of overloading the model, however, including a thoughtful mention of tradeoffs between **minimality** and **power**. In this section we give a brief recapitulation of the examples, together with our analyses.

3.1 Camps Redux

In this portion of his paper, Badia presents Camps' illustrations and conclusions, which he accepts. We've already discussed this.

3.2 Commutativity in ERD's

In mathematical contexts, we call a diagram commutative [14] if all different routes from a common source to a common destination are equivalent. In Figure 4, from Badia's paper (there called Figure 1), there are two different ways to navigate from **Course** to **Department**: directly, or via the **Teacher** entity. To say that this particular diagram commutes, then, is to say that for each course, its instructor must be a faculty member of the department that offers it. Again, there is a SQL construct for indicating this. Although Badia doesn't use the term, his point here is that there is no mechanism for ERD's to indicate a commutativity constraint. This is correct, of course. Consider the case of representing this kind of multi-relation constraints in the diagram with over just 50 entities and relationships, which are quite common in real-world applications. We believe, therefore, that this kind of a multi-relation constraint is better to be specified as a textual or a language-oriented syntax, such as OCL [17], rather than at a diagram level. In this way, a diagram can clearly deliver its major semantics without incurring visual overload and clutter.

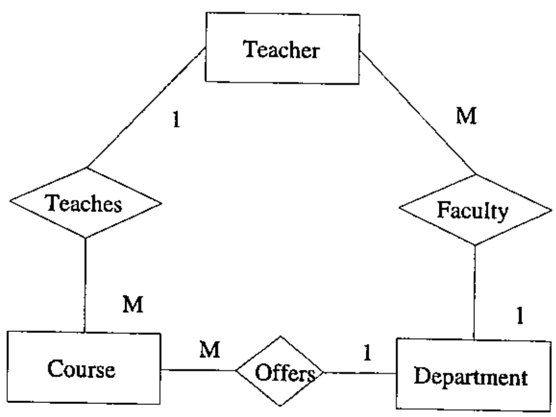

Fig. 4. An example of multi-paths between two entities (from Figure 1 in [2])

6/12

In certain limited situations [8] the **Offers** relationship might be superfluous and recovered by composing the other two relationships (or, in the relational database schema, by performing the appropriate joins). We would need to be careful about dropping **Offers**, however. For example, if a particular course were at present unstaffed, then the **Teaches** link would be broken. This is the case when **Course** entity has partial (optional) participation to **Department** entity. Without an explicit **Offers** instance, we wouldn't know which department offers the course. This is an example of a **chasm trap** which requires an explicit **Offers** relationship [6]. Another case where we couldn't rely on merely dropping one of the relationship links would arise if a commutative diagram involved the composition of two relationships in each path; then we would surely need to retain them both and to implement the constraint explicitly.

We note that allowing cycles and redundancies in ERD's has been a topic of research in the past. Atzeni and Parker [1] advise against it; Markowitz and Shoshani [15] feel that it is not harmful if it is done right. Dullea and Song [8, 9] provide a complete analysis of redundant relationships in cyclic ERD's. Their decision rules on redundant relationships are based on both maximum and minimum cardinality constraints.

3.3 Acyclicity of a Recursive Closure

Next, Badia considers the recursive relationship **ManagerOf** (on an **Employee** entity). He would like to accommodate the hierarchical property that nobody can be an indirect manager of oneself. Again, we agree with this observation but can't comment on how desirable such an ER feature would be at a diagram level. Badia points out that this is a problem even at the level of the relational database, although some Oracle releases can now accommodate the constraint.

3.4 Fan Traps

At this point the author brings Figure 5 (adapted from [6], where it appears as Figure 11.19(a); for Badia it is Figure 2) to our attention. (The original figure uses the "Merise," or "look here" approach [17]; we've modified it to make it consistent with the other figures in this paper.) The problem, called a **fan trap** arises when one attempts to enforce a constraint that a staff person must work in a branch operated by her/his division. This ER anomaly percolates to the relational schemas as well. Further, if one attempts to patch things up by including a third binary link, between **Staff** and **Branch**, then one is faced with the commutativity dilemma of Section 3.2. In general fan traps arise when there are two 1:N relationships from a common entity type to two different destinations. The two typical solutions for fan traps are either to add a third relationship between the two many-side entities or rearrange the entities to make the connection unambiguous. The problem in Figure 5 here is simply caused by an incorrect ERD and can be resolved by rearranging entities as shown in Figure 6. Figure 6 avoids the difficulties at both the ER and relational levels. In fact, this fix is even exhibited in the Connolly source itself. We note that the chasm trap discussed in Section 3.2 and the fan trap are commonly called **connection traps** [6] which make the connection between two entities separated by the third entity ambiguous.

7/12

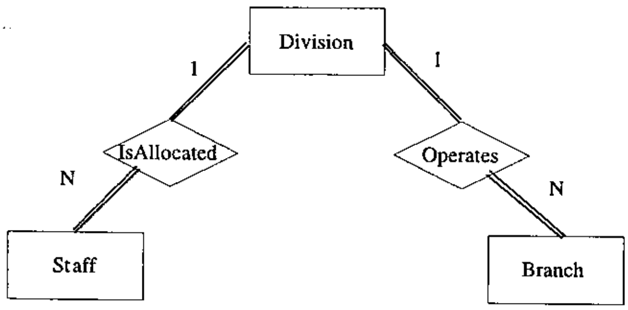

Fig. 5. A semantically wrong ERD with a fan trap (from Figure 2 in [2] and Figure 11.19(a) from [6])

Fig. 6. A correct ERD of Figure 5, after rearranging entities

3.5 Temporal Considerations

Here Badia looks at a **Works-in** relationship, M:N between **Employee** and **Project**, with attributes **start-date** and **end-date**. A diagram for this might look something like Figure 7b; for the purposes of clarity, most attributes have been omitted. Baida states that the rule that *even though en employee may work in many projects, an employee may not work in two projects at the same time* may not be represented in an ERD. It appears impossible to express the rule, although the relationship is indeed M:N. But wouldn't this problem be solved by creating a third entity type, **TimePeriod**, with the two date attributes as its composite key, and letting **Works-in** be ternary? The new relationship would be M:N:1, as indicated in Figure 7c, with the 1 on the **Project** node, of course. In figures of 7a through 7d, we show several variations of this case related to capturing the history of works-in relationships and the above constraint. We'll comment additionally on this in Section 4.

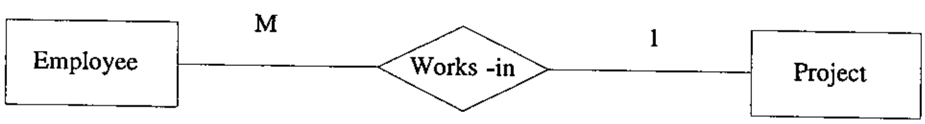

Fig. 7a. An employee may work in only one project and each project can have many employees. The diagram already assumes that an employee must work for only one project at a time. This diagram is **not** intended to capture any history of *works-in* relationship

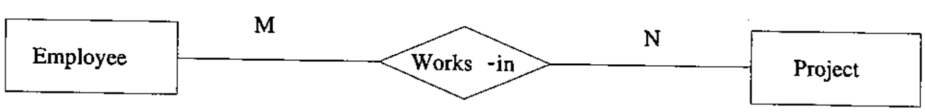

Fig. 7b. An employee may work in many projects and each project may have many employees. The diagram assumes that an employee may work for many projects at the same time. This diagram is also **not** intended to capture any history of *works-in* relationship

8/12

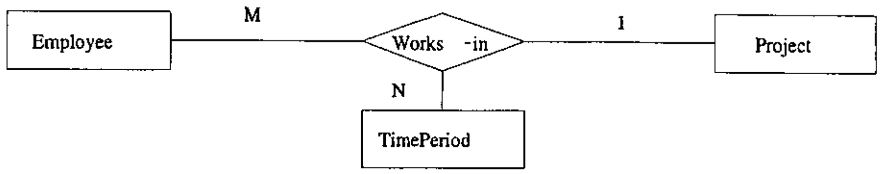

Fig. 7c. An employee may work in only one project at a time. This diagram can capture a history of *works-in* relationship of an employee for projects and still satisfies the constraint that an employee may work in only one project at a time

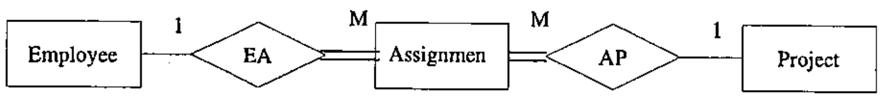

Fig. 7d. In Figure 7.c, if entity TimePeriod is not easily materialized, we can reify the relationship Works-in to an intersection entity. This diagram can capture the history of works-in relationship, but does not satisfy the constraint that an employee may work in only one project

3.6 Range Constraints

While the five previous cases exemplify what Badia calls **relationship constraints**, this one is an **attribute constraint**. The example given uses the following two tables:

Employee (employee_id, rank_id, salary, ...)
Rank (rank_id, max_salary, min_salary)

The stated problem is that the ERD that represents the above schema cannot express the fact that the salary of an employee must be within the range determined by his or her rank. Indeed, in order to enforce this constraint, explicit SQL code must be generated. Baida correctly sates that the absence of information at the attribute level is a limitation and cause difficulty in solving semantic heterogeneity. We believe, however, that information and constraints at the attribute level could be expressed at the data dictionary level or in a separate low level diagram below the ERD level. Again, this will keep an ERD as a *conceptual model* at enterprise level without too much clutter. Consider the complexity of representing attribute constraints in ERDs for real-world applications that have over 50 entities and several hundreds of attributes. The use of a CASE tool that supports a conceptual ERD with its any low level diagram for attributes and/or its associated data dictionary should be a right direction for this problem.

4 General Cardinality Constraints

While on the whole, as indicated above, we feel many of the alleged shortcomings of the ER model claimed in recent papers are not justified, some of those points have been well taken and are quite interesting. However, there is another important feature of conceptual design that we shall consider here, one that the ER model really does lack. In this section, we briefly discuss McAllister's general cardinality constraints [16] and their implications.

McAllister's setting is a general n-ary relationship **R**. In other words, R involves **n** different **roles**. This term is used, rather than **entity types**, since the entity types may not all be distinct. For example, a recursive relationship, while binary in the mathe-

9/12

matical sense, involves only a single entity type. Given two disjoint sets of roles A and B, McAllister defines **Cmax(A,B)** and **Cmin(A,B)** as follows: for a tuple <a>, with one component from each role in A, and a tuple <b>, with one component from each role in B, let us denote by <a,b> the tuple generated by the two sets of components; we recall that A and B are disjoint. Then Cmax(A,B) (respectively Cmin(A,B)) is the maximum allowable cardinality over all <a> of the set of tuples <b> such that <a,b>$\in$ $\pi_{A\cup B}$ (R). For example, consider the **Concession** relationship of Figure 1. Then to say that

Cmax({State, Product},{Dealer}) = 1 is to express the fact that {s-Id, p-Id} $\rightarrow$d-Id. And the condition **Cmin({Product},{State,Dealer}) = 1** is equivalent to the constraint that **Product** is total on **Concession**. Now, as we see from these examples, Cmax gives us information about functional dependencies and Cmin about participation constraints. When B is a singleton set and A its complement, this is sometimes called the "Chen" approach to cardinality [11] or "look across"; when A is a singleton set and B its complement, it is called the "Merise" approach [11] or "look here." All told, McAllister shows that there are 3^n-2^{n+1}+1 different combinations possible for A and B, where n is the number of different roles.

Clearly, given this explosive growth, it is impractical to include all possible cardinality constraints in a general ERD, although McAllister shows a tabular approach that works pretty well for ternary relationships. He shows further that there are many equalities and inequalities that must hold among the cardinalities, so that the entries in the table are far from independent. The question arises as to which cardinalities have the highest priorities and should thus appear in an ERD. It turns out that the Merise and Chen approaches give the same information in the binary case but not in the ternary one, which becomes the contentious case (n>3 is rare enough not to be a serious issue). In fact one finds both Chen [as in 10] and Merise [as in 3] systems in practice. In his article, Genova feels that UML [17] made the wrong choice by using the Chen method for its Cmin's, and he suggests that class diagrams include <u>both</u> sets of information (but only when either A or B is singleton). That does not seem likely to happen, though.

Still, consideration of these general cardinality constraints and McAllister's axioms comes in handy in a couple of the settings we have discussed. The general setting helps understand connections between, for example, ternary and related binary relationships as in Figure 2 and [12]. And it similarly sheds light on preservation (and loss) of information in Section 3.5 above, when a binary relationship is replaced by a ternary one. Finally, we believe that it also provides the deep structural information for describing the properties of decompositions of the associated relation schemas. It is therefore indisputable in our opinion that these general cardinality constraints do much to describe the fundamental structure of a relationship in the ER model; only portions of which, like the tip of an iceberg, are currently visible in a typical ERD. And yet we are not claiming that such information should routinely be included in the model.

5 Conclusion

We have reviewed recent literature ([4] and [2]) that illustrate through some interesting examples areas of conceptual database design that are not accommodated suffi-

10/12

ciently at the present time by the Entity-Relationship model. However, some of these examples seem not to hold up under scrutiny.

Capabilities that the model does indeed lack are constraints on commutative diagrams (Section 3.2 above), recursive closures (3.3), and some range conditions (3.6) as pointed out by Badia. Another major conceptual modeling tool missing in the ER model is that of general cardinality constraints [16]. These constraints are the deep structure that underlies such more visible behavior as constraining and related relationships, Chen and Merise cardinality constraints, functional dependencies and decompositions, and participation constraints. How many of these missing features should actually be incorporated into the ER model is pretty much a question of triage, of weighing the benefits of a feature against the danger of circuit overload.

We believe that some complex constraints such as multi-relation constraint are better to be represented as a textual or a language-oriented syntax, such as OCL [17], rather than at the ER diagram level. We also believe that information and constraints at the attribute level could be expressed at the data dictionary level or in a separate low level diagram below the ERD level. In these ways, we will keep an ERD as a *conceptual model* at enterprise level to deliver major semantics without visual overload and too much clutter. Consider the complexity of an ERD for a real-world application that has over 50 entities and hundreds of attributes and representing all those complex multi-relation and attribute constraints in the ERD. The use of a CASE tool that supports a conceptual ERD with its any low level diagram for attributes and/or its associated data dictionary should be a right direction for this problem.

We note that we do not claim that some research topics suggested by Baida, such as relationships over relationships and attributes over attributes, are not interesting or worthy. Research in those topics would bring interesting new insights and powerful ways of representing complex semantics. What we claim here is that the ERD itself has much value as it is now, especially for relational applications, where all the examples of Baida indicate. We believe, however, that extending the ER model to support new application semantics such as biological applications should be encouraged.

The "D" in ERD connotes to many researchers and practitioners the simplicity and power of communication that account for the model's popularity. Indeed, as the Entity-Relationship model nears its 30th birthday, we find its robustness remarkable.

References

1. 1.Atzeni, P. and Parker, D.S., "Assumptions in relational database theory", in *Proceedings of the 1st ACM Symposium on Principles of Database Systems*, March 1982.
2. Badia, A. "Entity-Relationship Modeling Revisited", *SIGMOD Record*, 33(1), March 2004, pp. 77-82.
3. Batini, C., Ceri, S., and Navathe, S., *Conceptual Database Design*, Benjamin/Cummings, 1992.
4. Camps Paré, R. "From Ternary Relationship to Relational Tables: A Case against Common Beliefs", *SIGMOD Record*, 31(20), June 2002, pp. 46-49.
5. Chen, P. "The Entity-Relationship Model – towards a Unified View of Data", *ACM Transactions on Database Systems*, 1(1), 1976, pp. 9-36.
6. Connolly, T. and Begg, C., *Database Systems*, 3d Edition, Addison-Wesley, 2002.
7. Dietrich, S. and Urban, S., *Beyond Relational Databases*, Prentice-Hall, to appear.
8. Dullea, J. and Song, I.-Y., "An Analysis of Cardinality Constraints in Redundant Relationships," *in Proceedings of Sixth International Conferences on Information and Knowledge Management (CIKM97)*, Las Vegas, Nevada, USA, Nov. 10-14, 1997, pp. 270-277.

11/12

9. Dullea, J., Song, I.-Y., and Lamprou, I., "An Analysis of Structural Validity in Entity-Relationship Modeling," *Data and Knowledge Engineering*, 47(3), 2003, pp. 167-205.
10. Elmasri, R. and Navathe, S.B., *Fundamentals of Database Systems*, 4th Ed., Addison-Wesley, 2003.
11. Genova, G., Llorenz, J., and Martinez, P., "The meaning of multiplicity of n-ary associations in UML", Journal of Software and Systems Modeling, 1(2), 2002.
12. Jones, T. and Song, I.-Y., "Analysis of binary/ternary cardinality combinations in entity-relationship modeling", *Data & Knowledge Engineering*, 19(1), 1996, pp. 39-64.
13. Jones, T. and Song, I.-Y., "Binary Equivalents of Ternary Relationships in Entity-Relationship Modeling: a Logical Decomposition Approach." *Journal of Database Management*, 11(2), 2000, (April-June, 2000), pp. 12-19.
14. MacLane, S., *Categories for the Working Mathematician*, Springer-Verlag, 1971.
15. Markowitz, V. and Shoshani, A., "Representing Extended Entity-Relationship Structures in Relational Databases: A Modular Approach", *ACM Transactions on Database Systems*, 17(3), 1992, pp. 423-464.
16. McAllister, A., "Complete rules for n-ary relationship cardinality constraints", *Data & Knowledge Engineering*, 27, 1998, pp. 255-288.
17. Rumbaugh, J., Jacobson, I., and Booch, G., *The Unified Modeling Language Reference Manual*, Addison-Wesley, 1999.
18. Song, I.-Y., Evans, M., and Park, E.K., "A Comparative Analysis of Entity-Relationship Diagrams," *Journal of Computer and Software Engineering*, 3(4) (1995), pp. 427-459.
19. Teorey, T., *Database Modeling & Design*, 3d Edition, Morgan Kaufmann, 1999.

Ph.D. Entrance Exam- IT

Please read the attached paper and answer the following questions. Note that the time is limited. Plan your time wisely so that you would have time to answer the questions below. Try to find out the proper answer for the respective question. We expect that you would use your own words to describe/explain, instead of copying from the paper.

1. (10%) What is the problem that the paper proposes to solve?
2. (20%) Describe the proposed distributed stream management infrastructure. Describe the functionality of each layer as well.
3. (20%) How can the proposed network partitioning method be applied to other research issues?

Resource-Aware Distributed Stream Management using Dynamic Overlays

Vibhore Kumar, Brian F Cooper, Zhongtang Cai, Greg Eisenhauer, Karsten Schwan
College of Computing, Georgia Institute of Technology
801 Atlantic Drive, Atlanta, GA 30332-0280
{vibhore, cooperb, ztcai, eisen, schwan}@cc.gatech.edu

Abstract

We consider distributed applications that continuously stream data across the network, where data needs to be aggregated and processed to produce a 'useful' stream of updates. Centralized approaches to performing data aggregation suffer from high communication overheads, lack of scalability, and unpredictably high processing workloads at central servers. This paper describes a scalable and efficient solution to distributed stream management based on (1) resource-awareness, which is middleware-level knowledge of underlying network and processing resources, (2) overlay-based in-network data aggregation, and (3) high-level programming constructs to describe data-flow graphs for composing useful streams. Technical contributions include a novel algorithm based on resource-aware network partitioning to support dynamic deployment of data-flow graph components across the network, where efficiency of the deployed overlay is maintained by making use of partition-level resource-awareness. Contributions also include efficient middleware-based support for component deployment, utilizing runtime code generation rather than interpretation techniques, thereby addressing both high performance and resource-constrained applications. Finally, simulation experiments and benchmarks attained with actual operational data corroborate this paper's claims.

1. Introduction

Many emerging distributed applications must cope with a critical issue: how to efficiently aggregate, use, and make sense of the enormous amounts of data that is generated by these applications. Examples include sensor systems [1], distributed scientific processes like SkyServer [2], operational information systems used by large corporations [3], and others. Middleware initiatives for such applications include publish/subscribe systems like IBM's Gryphon [4] project or related academic endeavors [5], or commercial infrastructures based on web services, based on technologies like TPF, or using in-house solutions. However, middleware that relies on centralized approaches to data aggregation suffers from high communication overheads, lack of scalability, and unpredictably high processing workloads at central servers.

Our solution is to use in-network aggregation to reduce the load problems encountered in centralized approaches. This approach exploits the fact that data in these applications is usually routed using overlay networks, such that updates from distributed data sources arrive at their destination after traversing a number of intermediate overlay nodes. Each intermediate node can contribute some of its cycles towards processing of the updates it is forwarding, the resulting advantage being the distribution of processing workload and a possible reduction in communication overhead involved in transmitting data updates.

In this paper, we examine how to construct a distributed system for processing and aggregating streams of data. However, in order to set up such a system with nodes ready to contribute their resources for data processing, we must address several challenges, including:

- *Ease of Deployment* – provide a simple interface for composing new streams from existing streaming data and also support run-time modifications to stream composition conditions.
- *Scalability* – there may be hundreds of streaming data sources, and the system should be incrementally scalable without significant overhead or effort.
- *Heterogeneity* – streaming data arrives at the sink after traversing a possibly heterogeneous set of nodes, which means that the system should support in-network processing of the streams at any node despite varying resource capabilities and operating environments.
- *Dynamism* – should automatically reconfigure to deal with changes in network conditions, node overloads and changes in data stream rates.

【背面還有試題】

- *Performance* – since updates arrive at a very high rate, the infrastructure should not impose large overheads when aggregating and processing the updates.

We are implementing a Distributed Stream Management Infrastructure (DSMI) to compose new data streams by aggregating and processing existing streaming data originating at distributed locations. The system supports a SQL-like language to describe the data-flow graph for producing the new, transformed stream from existing data streams. The language allows users to refer to any stream originating in the system and supports attribute selection and join operations as in traditional databases. A resource-aware network-partitioning algorithm, described later, is used to assign operators from the flow graph to the underlying network nodes. The infrastructure relies on ECho [5], a pub-sub middleware developed at Georgia Tech, to deploy the data-operators for processing and forwarding the streaming updates in a heterogeneous environment. Automatic reconfiguration of stream overlays is achieved by coupling the resource information collected from participating hosts with the Proactive Directory Service [7] (PDS), which is a subscription-based monitoring tool also developed by our group.

1.1. Example: Operational Information System

An operational information system (OIS) [6] is a large-scale, distributed system that provides continuous support for a company or organization's daily operations. One example of such a system we have been studying is the OIS run by Delta Air Lines, which provides the company with up-to-date information about all of their flight operations, including crews, passengers and baggage. Delta's OIS combines three different sets of functionality:

- *Continuous data capture* – for information like crew dispositions, passengers, airplanes and their current locations determined from FAA radar data.
- *Continuous status updates* – for low-end devices like airport flight displays, for the PCs used by gate agents, and even for large databases in which operational state changes are recorded for logging purposes.
- *Responses to client requests* – an OIS not only captures data and updates/distributes operational state, but it must also respond to explicit client requests such as pulling up information regarding bookings of a particular passenger. Certain clients may also generate additional state updates, such as changes in flights, crews or passengers.

The key problems addressed by this paper are to reduce communication overheads, by selectively streaming the events; distributing the processing workload by using the computing resources spread across the organization; and implementing easy to use high-level language constructs for specifying new data-flow graphs.

1.2. Related Work

The stream management infrastructure we have implemented is very closely associated with topics of interest to the middleware community, and to those interested in large-scale distributed data management.

Stream Processing & Distributed Databases

Data-stream processing has recently been an area of tremendous activity for database researchers; several groups such as STREAM [10] at Stanford, Aurora [11] at Brown and MIT, and Infopipes [12] at Georgia Tech have been working to formalize and implement the concepts for data-stream processing. Most of these efforts have commonly assumed an on-line warehousing model where all source streams are routed to a central site where they are processed. There have also been some preliminary proposals that extend the single-site model to multi-site, distributed models and environments [13, 14]. Our work is also a step in this general direction. Of particular mention is the work by Madden et al. [15] that demonstrates the advantage of in-network data-aggregation in a wireless multi-hop sensor network.

Distributed query optimization deals with site selection for the various operators and has been explored in great detail in the context of distributed and federated databases [16, 17]. However, these systems do not deal with streaming queries over streaming data, which present new challenges, especially in dealing with resource limitations.

Pub-Sub Middleware

Pub-sub middleware like IBM's Gryphon [4], ECho [5], ARMADA [18] and more recently Hermes [19] have well established themselves as messaging middleware. Such systems address issues like determining who needs what data, building scalable messaging systems and simplifying the development of messaging applications. We believe that our work is the necessary next step that utilizes the middleware to provide high-level programming constructs to describe resource-aware and 'useful' data-flows.

Network Partitioning & Overlay Networks

Distribution and allocation of tasks has been a long studied topic in distributed environments. Architectural initiatives tailored for large-scale applications include SkyServer [2], enterprise management solutions [3] and grid computing efforts [29]. These applications perform task allocation to servers much in the same way as we recursively map operators to nodes. However, a high-level construct for describing the data-flow and run-time re-assignment of operators based on an application-driven utility distinguishes our infrastructure.

Overlay networks [20, 21] focus on addressing scalability and fault tolerance issues that arise in large-scale content dissemination. The intermediate routers in overlay network perform certain operations that can be viewed as in-network data-aggregation but are severely restricted in their functionality. The advantages of using our infrastructure are two-fold; first its ability to deploy operators at any node in the network, and second is the ease with which these operators can be expressed. There has also been some work on resource-aware overlays [27], which is similar to resource-aware reconfiguration of the stream overlay in our infrastructure. In our case reconfiguration is very closely associated with the application level data requirements.

1.3. Roadmap

The remainder of this paper is organized as follows. In Section 2, we discuss the design of the basic components of the infrastructure, explaining its layered architecture and a brief description of the layers' functionality. Section 3 describes the data-flow graph deployment problem in detail, followed by a brief description of the network hierarchy and its use in deployment and maintenance of the stream overlay. Implementation issues for the infrastructure are discussed in Section 4. Section 5 presents an experimental evaluation of the proposed deployment algorithm, including results that were obtained using real enterprise data. Finally we conclude in Section 6 with a discussion of possible future directions.

Application Layer: Data-Flow Parser	
pub-sub Middleware **ECho**	***Resource Monitoring*** **PDS**
Underlay Layer: Network Partitioning	

Figure 1. Three layered architecture of the DSMI

2. Software Architecture

Our distributed stream management infrastructure (DSMI) is broadly composed of three layers as shown in Figure 1: (1) the Application Layer is responsible for accepting and parsing the data composition requests and constructing the data-flow graph, (2) the Middleware Layer consists of the ECho middleware and the PDS resource-monitoring infrastructure for deployment and maintenance of the stream overlay, and (3) the Underlay Layer organizes the nodes into hierarchical partitions that are used by the deployment infrastructure. The following subsections briefly describe these three layers.

2.1 Application Layer: Data-Flow Graph

Data flows are specified with our data-flow specification language. It closely follows the semantics and syntax of the SQL database language. The general syntax of our language is specified as follows –

```
STREAM <attribute1> [<,attribute2> [<,attribute3> ...]]
FROM <stream1> [<,stream2> [<,stream3> ...]]
[WHEN <condition1> [<conjuction> <condition2>[...]]];
```

In the data-flow specification language, the attribute list mentioned after the STREAM clause describes which components of each update are to be selected, the stream list following the FROM clause identifies the data stream sources, and finally, predicates are specified using the WHEN clause. Each stream in the infrastructure is addressable using the syntax `source_name.stream_name`. Likewise, an attribute in the stream is addressable using `source_name.stream_name.attribute`. Our language supports in-line operations on the attributes that are specified as `operator(attribute_list [,parameter_list])`, where examples of such operations include `SUM`, `MAX`, `MIN`, `AVG`, `PRECISION`, etc. The system also provides the facility to extend this feature by adding user-defined operators. An example data-flow description is shown in Figure 3.

The data-flow description is compiled to produce a data-flow graph. This graph consists of a set of operators to perform data transformations, as well as edges representing data streaming between operators. This graph is deployed in the network by assigning operators to network nodes.

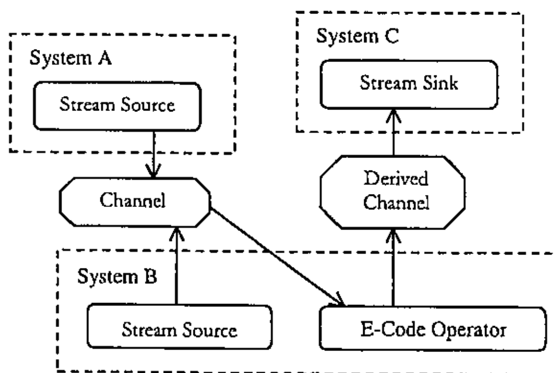

Figure 2. The ECho Framework

2.2. Middleware Layer: ECho and PDS

The Middleware Layer supports the deployment and reconfiguration of the data-flow overlay. This support is provided by two components: ECho and PDS.

The ECho framework is a publish/subscribe middleware system that uses channel-based subscription (similar to CORBA). ECho streams data over *stream channels*, which implement the edges between operators in the data-flow graph. The stream channels in our framework are not centralized; instead, they are lightweight distributed virtual entities managing data transmitted by middleware components at stream sources and sinks. An example system is shown in Figure 2. The traffic for individual channels is multiplexed over shared communication links by aggregating the traffic of multiple streams into a single stream linking the two communicating addresses.

We follow the semantics of a publish-subscribe system in order to ensure that multiple sinks can subscribe/unsubscribe to a stream channel depending on their requirements, and that the channels survive even when there are no subscribers (although in that case no actual data is streamed). The publish-subscribe system also proves useful when many sinks have similar data filtering needs; in such a scenario, a single channel derived using a data transformation operator can fill the needs of all the sinks.

The data-operator in our infrastructure is typically a snippet of code written in a portable subset of C called "E-Code". This snippet is transported as a string to the node where it has to be deployed. At the target-node, the code snippet is parsed, and native code is generated. The implicit context in which the code is executed is a function declaration of the form:

```
int operator( <input type> in, <output type> out)
```

A return value of 1 causes the update to be submitted, while a return value of 0 causes the update to be discarded. The function body may also modify the input before copying it to the output. New flow-graphs may use the streams from existing operators, or may cause operators to be created or updated to stream additional relevant data if necessary.

Network-wide resource availability information is managed by the Proactive Directory Service (PDS). This information allows us to dynamically reconfigure the data-flow deployment in response to changing resource conditions. PDS is an efficient and scalable information repository with an interface that includes a proactive, push-based access mode. Through this interface, PDS clients can learn about objects (or types of objects) inserted in/removed from their environment and about changes to pre-existing objects. The infrastructure uses PDS objects to receive resource updates from the system when operating conditions change.

2.3 Underlay Layer: Network Partitioning

This layer is responsible for maintaining a hierarchy of physical nodes in order to cluster nodes that are "close" in the network sense, based on measures like end-to-end delay, bandwidth or inter-node traversal cost (a combination of bandwidth and delay). The hierarchy is

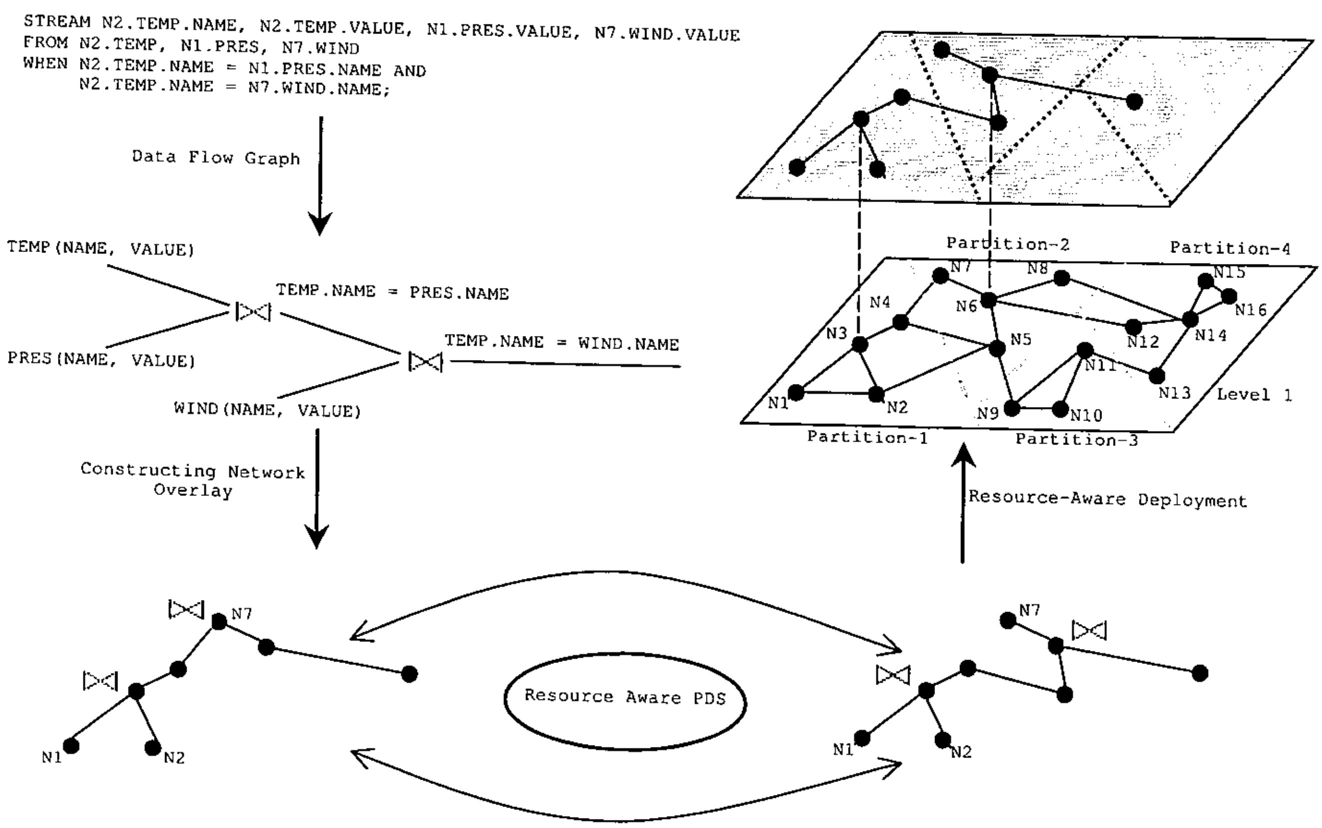

Figure 3. Steps involved in deploying the stream overlay

used for network-aware deployment of the data-flow graph. Each node in a cluster knows about the costs of paths between each pair of nodes in the cluster. A node is chosen from each cluster to act as the coordinator for this cluster in the next level of the hierarchy. Like the physical nodes in the first level of hierarchy, the coordinator nodes can also be clustered to add another level in the hierarchy; similar to the initial level all the coordinators at a particular level know about average min cost path to the other coordinator nodes that fall in the same partition at that level. An example is shown in Figure 4.

The advantage of organizing nodes in a hierarchy is that it simplifies maintenance of the clustering structure, and provides a simplified abstraction of the underlying network to the upper layers. Then, we can subdivide the data-flow graph to the individual clusters for further deployment. In order to scalably cluster nodes, we bound the amount of non-local information maintained by nodes by limiting the number of nodes that are allowed per cluster.

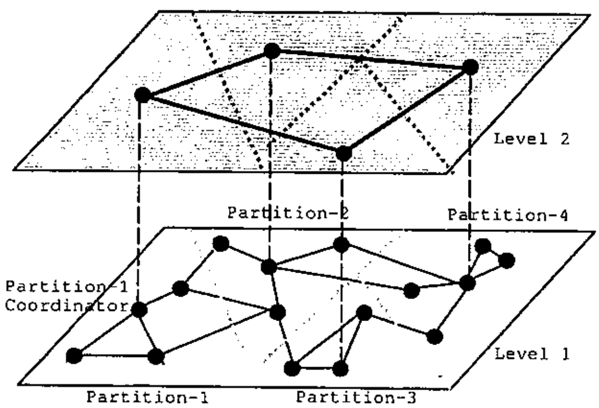

Figure 4. Hierarchical Network Partitioning

3 Deployment & Dynamic Reconfiguration

This section formally describes the data-flow graph deployment problem and then presents a highly scalable distributed algorithm that can be used to obtain an efficient solution to this problem. Then, we extend the deployment algorithm by incorporating resource-awareness.

3.1 Problem Statement

We consider the underlying network as a graph $N(V_n, E_n)$, where vertices V_n represent the actual physical nodes and the network connections between the nodes are represented by the edges E_n. We further associate each edge e_{ni} with a cost c_i that represents the application-oriented cost of traversing the corresponding network link. The data-flow graph derived from the SQL-like description is similarly represented as a graph $G(V_g, E_g)$ with each vertex in V_g representing a source-node, a sink-node or an operator i.e.

$$V_g = V_{g\text{-}sources} \cup V_{g\text{-}sink} \cup V_{g\text{-}operators}$$

$V_{g\text{-}sources}$ is the set of stream sources for a particular data-flow graph and each source has a static association with a vertex in graph N. Source vertices have an associated update-rate. $V_{g\text{-}sink}$ is the sink for the resulting stream of updates and it also has a static association with a vertex in graph N. $V_{g\text{-}operators}$ is the set of operators that can be dynamically associated with any vertex in graph N. Each operator vertex is characterized by a resolution factor, which represents the increase or decrease in the data flow caused by the operator. In general, join operators, which combine multiple streams, increase the amount of data-flow; while select operations, which filter data from a stream, result in a corresponding decrease. The edges in the data-flow graph may span multiple intermediate edges and nodes in the underlying network graph.

We want to produce a mapping M, which assigns each $v_{gj} \in V_{g\text{-}operators}$ to a $v_{ni} \in V_n$. Thus, M implies a corresponding mapping of edges in G to edges in N, such that each edge $e_{gj\text{-}k}$ between operators v_{gj} and v_{gk} is mapped to the network edges along the lowest cost path between the network nodes that v_{gj} and v_{gk} are assigned to. We define $cost(M)$ as the sum of the costs of the network edges mapped to the edges in the data flow graph:

$$cost(M) = \sum_{e_{ni} \in M} cost(e_{ni})$$

For example, consider a cost function that measures the end-to-end delay. If e_{gk} is determined by vertices v_{gi} and v_{gj}, which in turn are assigned to vertices v_{ni} and v_{nj} of the network graph N, then the cost corresponding to edge e_g is the delay along the shortest path between the vertices v_{ni} and v_{nj}.

The problem is to construct the lowest cost mapping M between the edges E_g in G to edges E_n in N.

3.2 Distributed Deployment Algorithm

Now, we present a distributed algorithm for deploying the dataflow graph in the network. In a trivial scenario we could have a central planner assign operators to network nodes, but this approach will obviously not scale for very large networks, and the planner can become a central point of failure. Our partitioning-based approach deploys the data-flow graph in a more decentralized way. In particular, nodes in the network self-organize into a network-aware set of clusters, such that nodes in the same cluster have low latency. Then, we can use this partitioned structure to deploy the data-flow graph in a network-aware way, without having full knowledge of the delay between all pairs of network nodes.

The result is that an efficient mapping M is constructed recursively, using the hierarchical structure of the underlay-layer. This mapping may not be optimal, since our approach trades guaranteed optimality for scalable deployment. However, since the deployment is network-aware, the mapping should have low cost. Experiments presented in Section 5 demonstrate that our algorithm produces efficient deployments.

We now formalize the partitioning scheme described in Section 2.3. Let

n^i_{total} = total nodes at level i of the hierarchy

$n_{critical}$ = maximum number of nodes per partition

v^i_{nj} = coordinator node for node v_{nj} at level i

Note that $v^0_{nj} = v_{nj}$ and that all the participants of a partition know about minimum cost path to all other nodes in the same partition. We bound the amount of path information that each node has to keep by limiting the size of the cluster using $n_{critical}$. A certain level i in the hierarchy is partitioned when $n^i_{total} > n_{critical}$. We consider the physical nodes to be located at level 1 of the partition hierarchy and actual network values are used to partition nodes at this level. For any other level i in the hierarchy the average inter-partition cost (i.e. end-to-end delay, bandwidth, etc.) from level i-1 are used for partitioning the coordinator nodes from the level i-1. The approximate cost between any two vertices v_{nj} and v_{nk} at any level i in the hierarchy can be determined using the following equations:

$$cost^i(v_{nj}, v_{nk}) = \begin{cases} cost(v^{i-1}_{nj}, v^{i-1}_{nk}) & for\, v^{i-1}_{nj} \neq v^{i-1}_{nk} \\ 0 & for\, v^{i-1}_{nj} = v^{i-1}_{nk} \end{cases}$$

and

$$cost(v_{nj}, v_{nk}) = cost^l(v_{nj}, v_{nk}) \mid v^{l-1}_{nj} \neq v^{l-1}_{nk} \wedge v^l_{nj} = v^l_{nk}$$

In simple words, the cost at level i between any two vertices v_{nj} and v_{nk} of N is 0 if the vertices have the same coordinator at level i-1, otherwise it is equal to the cost at some level l where the vertices have the same coordinator and do not share the same coordinator at level l-1.

The distributed deployment algorithm works as follows, the given data-flow graph $G(V_g, E_g)$ is submitted as input to the top-level (say level t) coordinators. We construct a set of possible node assignments at level t by exhaustively mapping all the vertices $V_{g\text{-}operators}$ in V_g to the nodes at this level. The cost for each assignment is calculated using the algorithm shown in Figure 5 and the assignment with lowest cost is chosen. This partitions the graph G into a number of sub-graphs each allocated to a node at level t and therefore to a corresponding cluster at level t-1. The sub-graphs are then again deployed in a similar manner at level t-1. This process continues till we reach level 1, which is the level at which all the physical nodes reside.

3.3 Reconfiguration

The overlay reconfiguration process takes advantage of two important features of our infrastructure; (1) that the nodes reside in clusters and (2) that only intra-cluster minimum cost analysis is required. These features allow us to limit the reconfiguration to within the cluster boundaries, which in turn makes reconfiguration a low-overhead process. An overlay can be reconfigured in response to a variety of events, which are reported to the first-level cluster-coordinators by the PDS. These events include change in network delays, change in available bandwidth, change in data-operator behavior (we call this operator profiling), available processing capability, etc. Since it is impractical to respond to all such events reported by the PDS, we set thresholds that should be reached to trigger a reconfiguration. For example, a cluster-coordinator may recalculate the minimum cost paths and redeploy the assigned sub-graphs when more than half the links in the cluster have reported change in end-to-end delay. However, setting such thresholds depends on the application-level requirement for resource-awareness. In ongoing work we are developing a closer integration between the application-level requirements and the reconfiguration framework.

Note that reconfigurations are not lossless in terms of updates and some updates and state maybe lost during the process. This is acceptable for most of the streaming applications, which are able to tolerate some level of approximation and loss. However, as part of the ongoing work we are trying to model reconfiguration as a database-style transaction in order to achieve losslessness.

```
graphCost(Node n, Level l){
  if(n->left == null && n->right == null)
    return 0;
  if(n->left != null)
    cl = pathCost(n->node[l], n->left->node[l])
         + graphCost(n->left, l);
  if(n->right != null)
    cr = pathCost(n->node[l], n->right->node[l])
         + graphCost(n->right, l);
 return cl + cr;
}
```

Figure 5. Algorithm for calculating graph cost at a level

3.4 Advantages of the DSMI Approach

DSMI is an infrastructure for distributed processing of data streams. Its main contribution is in composing and transforming data streams using a data flow graph that can be defined declaratively and deployed efficiently. We have already discussed the algorithm used for resource-aware deployment of the stream overlay. Following are the advantages of our architecture for stream management:

- *Online Predicate Rewriting* – the data-operators allow the stream sink to fine-tune their behavior by updating a remotely accessible data-structure associated with each operator. This is used to support dynamic modification of composition parameters and conditions.
- *Resource-Aware Reconfiguration* – once deployed, a stream overlay reacts to changes in operating environment at two levels: to respond to changes in local conditions, an intra-partition level reconfiguration is done, while an inter-partition level reconfiguration handles substantial changes in operating conditions.
- *Operator Duplication* – the system allows for duplication of a limited type of operators to achieve parallel processing of some rapid update streams in resource-constrained environments. This technique may result in some updates being re-ordered, so it is only applicable in certain scenarios.
- *Operator Migration* – the system allows the user to initiate operator reconfigurations to complement the automatic resource-aware reconfiguration being done by the system.
- *Embedded Portability* - the E-Code Language is a portable subset of C that includes enough functionality to implement stream operators. The E-Code compiler can be extended to include other language features and can be easily ported to new platforms. Therefore, the operators are easily portable to heterogeneous nodes.
- *Minimal Impact on Performance* - given that the data-operators are invoked for each update and are deployed on a remote machine, the performance of the system depends heavily on the performance of these operators. Dynamic native compilation of the data-operators reduces the overheads of update processing.

```
{
  // an example of customized computation
  int i, j;
  for(i = 0; i<256; i= i+2) {
    for(j = 0; j<256; j=j+2) {
      output.image[i/2][j/2] =
(input.image[i][j] + input.image[i+1][j+1] +
input.image[i][j+1] + input.image[i+1][j]) / 4 ;
    }
  }
  return 1;
}
```

Figure 6. Resolution Reduction Operator as E-Code

4 Implementation

DSMI has been implemented using C++ and is closely integrated with the ECho and the PDS modules developed by our group. The system is brought-up by specifying a set of initial nodes, and an instance of DSMI is started at a well-known port on each such node. A node can join or leave the infrastructure at a later point of time. The system maintains a distributed hash table to map the user specified unique node names to the IP-address for each node. We utilize user-specified names to facilitate the task of graph composition by users. The hierarchical node-partitioning module runs an iterative variant of the well-known k-means [28] clustering algorithm to partition nodes based on inter-node end-to-end delay values.

A user can create a stream schema at any node by using the CREATE STREAM command at the infrastructure prompt. Registering a schema causes an ECho typed-channel to be created that has the capability to carry any update, which conforms to the specified schema. The node can then start streaming data on this channel and other node can refer to this stream as `node_name.stream_name`.

When a stream composition request is submitted at any node in the infrastructure, it is parsed to create a data-flow graph. Each edge in the graph is mapped to an ECho channel, which is instantiated with appropriate data carrying capability. Each operator in the flow graph is either a pre-compiled routine (operators which are hard to express as E-Code) or an appropriate E-Code snippet. An example image resolution reduction snippet written in E-Code is shown in Figure 6. The flow-graph operator consists of one or two incoming channels, an outgoing channel and an operator routine. Operator information is specified in XML and sent to the node where the operator has to be instantiated. Since we are using a pub-sub middleware, instantiating an operator consists of becoming a subscriber to appropriate incoming channel(s), a publisher to the outgoing channel and starting the associated operator routine. The ease of operator deployment helps us to reduce the overhead during reconfigurations as no new channels are created; only the channel publisher and the subscriber change to reconfigure the overlay.

5 Experiments

We ran a set of experiments to evaluate the performance of our architecture. First, we ran microbenchmarks to examine specific features of our system. Then, we created an end-to-end setup for an application case study using real data from Delta Airlines' OIS. Our results show that our system is effective at deploying and reconfiguring data-flow graphs for distributed processing of streaming data.

5.1 Experimental setup

The GT-ITM internetwork topology generator [8] was used to generate a sample Internet topology for evaluating our deployment algorithm. This topology represents a distributed OIS scattered across several locations. Specifically, we use the transit-stub topology for the ns-2 simulation by including one transit domain that resembles the backbone Internet and four stub domains that connect to transit nodes via gateway nodes in the sub domains. Each stub domain has 32 nodes and the number of total

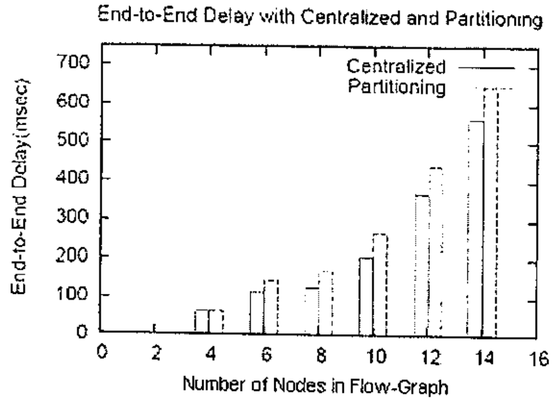

Figure 7. Comparison of end-to-end delay for centralized and partitioning approach

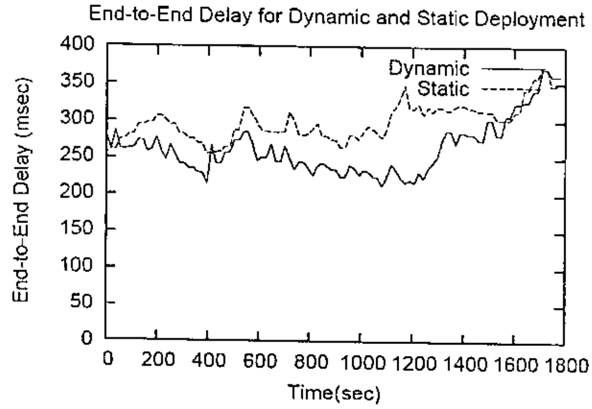

Figure 8. Variation of end-to-end delay with and without dynamic reconfiguration

transit nodes is 128. Links inside a stub domain are 100Mbps. Links connecting stub and transit domains, and links inside a transit domain are 622Mbps, resembling OC-12 lines. The traffic inside the topology was composed of 900 CBR connections between sub domain nodes generated by cmu-scen-gen [9]. The simulation was carried out for 1800 seconds and snapshots capturing end-to-end delay between directly connected nodes were taken every 5 seconds. These are then used as inputs for our distributed deployment algorithm.

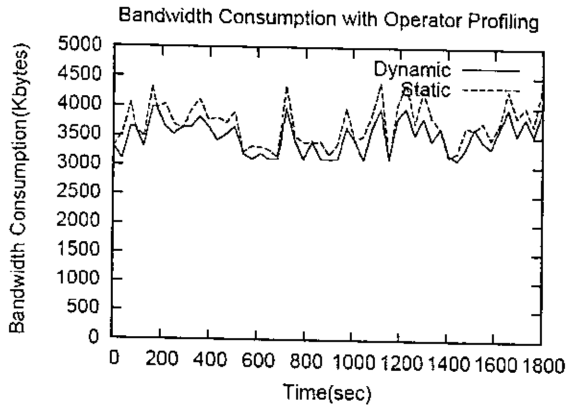

Figure 9. Variation in bandwidth consumption with dynamic reconfiguration using Operator Profiling

5.2 Microbenchmarks

The first experiment focused on comparing the cost of a deployed data-flow graph using the centralized model as opposed to the partitioning based approach used in our infrastructure. Since in centralized approach we assume that a single node knows about minimum cost paths to all other nodes, the centralized approach gives the optimal deployment solution. However, the deployment time taken by centralized approach increases exponentially with the number of nodes in the network. Figure 7 shows that although the partitioned-based approach is not optimal, the cost of the deployed flow graph is not much worse than the deployment in the centralized approach, and is thus suitable for most scenarios.

The next experiment was conducted to examine the effectiveness of dynamic reconfiguration in providing an efficient deployment. Figure 8 shows the variation of end-to-end delay for a 10-node data-flow graph with changing network conditions, as simulated by introducing cross-traffic. The performance with dynamic reconfiguration is clearly better than with static deployment. It may be noted that at some points, cost of the dynamically reconfigured flow-graph becomes more than that of the static deployment. This happens because the cost calculation algorithm used in our approach calculates the graph cost that is an approximation of the actual deployment cost. In some cases the approximation is inaccurate, causing the reconfiguration to make a poor choice. However, these instances are rare, and when they do occur, the cost of the dynamic deployment is not much worse than the static deployment. Moreover, for most of the time dynamic reconfiguration produces a lower cost deployment.

We also conducted experiments to compare the bandwidth consumption with and without dynamic reconfiguration. Each source was assumed to have a certain update rate of the form bytes/sec and each link was associated with a cost incurred per byte of data transferred using the link. Thus, at any point of time a deployed data graph has a cost, which is dependent on the links being used by the flow. We simulated a change in resolution factor (the ratio of the amount of data flowing out versus flowing in) for each operator in the flow graph and measured the corresponding bandwidth utilization with dynamic reconfiguration and static deployment. We notice that although dynamic reconfiguration helps in keeping the bandwidth consumption low, it does not offer very substantial gains; this is because when reconfiguration is driven by operator resolution it offers only a limited space for re-deployment.

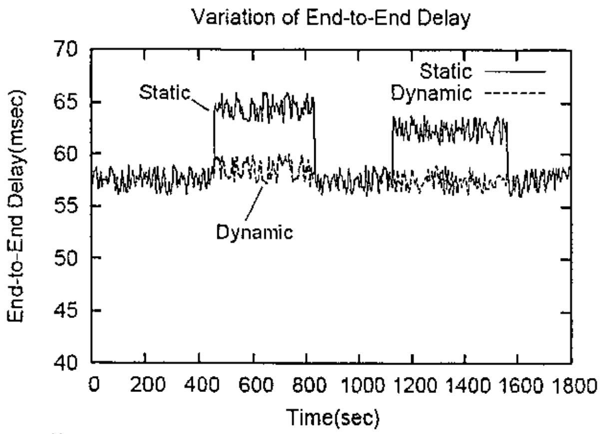

Figure 10. Variation of end-to-end delay for network perturbation and processor overload

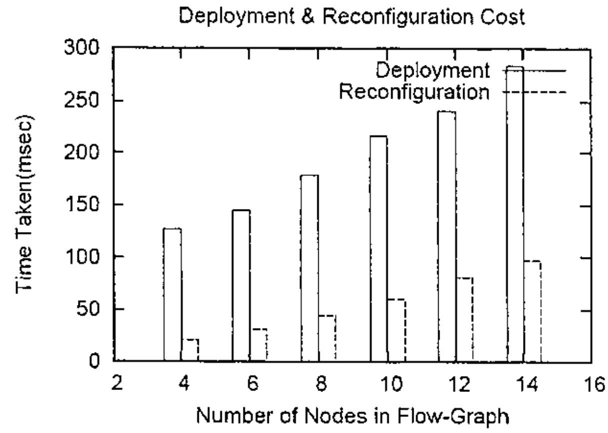

Figure 11. Comparison of deployment and reconfiguration cost on Emulab nodes

5.3 Application case study

The next set of experiment was conducted on Emulab [23] with real data from the Delta OIS combined with simulated streams for Weather and News. The experiment was designed to emulate 4 different airport locations. The inter-location delays were set to ~50ms while delays within an airport location were set to ~2ms. The emulation was conducted with 13 nodes (Pentium-III, 850Mhz, 512MB RAM, RedHat Linux 7.1) and each location had only limited nodes connected to external locations. The experiment was motivated by the requirement to feed overhead displays at airports with up-to-date information. The overhead displays periodically update the weather and news at 'destination' location and switch over to seating information for the aircraft at boarding gate. Other information displayed on such monitors includes names of wait-listed passengers, and current status of flight, etc. We deployed a flow graph with two operators, one for combining the weather and news information at destination and the other for selecting the appropriate flight data, which originates from a central location (Delta's TPF facility in this case).

The first experiment conducted on Emulab studied the behavior of system in case of network perturbation and then studied its response to processor overload. Once the data flow graph for providing an overhead display feed was deployed, we used iperf [24] to introduce traffic in some of the links used by the flow-graph. This is represented by the first spike in Figure 10. With dynamic reconfiguration the flow-graph responds well to the spike in traffic; in contrast, the statically deployed graph experiences an increased delay. The next spike is a result of an increased processing load at both the operator nodes. Again with dynamic reconfiguration we end with a better delay than the static deployment. Even with dynamic reconfiguration the end-to-end delay spikes, but the time before the deployment adjusts is so short (milliseconds) that the spike is effectively unnoticeable.

The next experiment was conducted to compare the time for initial deployment and reconfiguration. Figure 11 shows that the times are quite small; only a few hundred milliseconds in the worst case. The figure illustrates the advantage of using a pub-sub middleware for deploying the flow graph. The pub-sub channels have to be created only at the time of deployment; reconfiguration just involves a change in publisher and subscriber to this channel and is therefore even faster. It may also be noted that once the channels for the data-flow graph have been created, deployment is essentially a distributed process, which starts once the corresponding nodes receive the operator deployment messages. This makes deployment time to increase almost linearly with the number of nodes.

Table 1. Middleware: Send & Receive Costs

Msg Size (bytes)	Send Cost (ms)	Receive Cost (ms)
125	0.084	0.154
1250	0.090	0.194
12500	0.124	0.327

Middleware Microbenchmarks on Emulab

Table 1 gives a measure of the low send and receive overheads imposed by the middleware layer at the intermediate nodes using the above setup. Send-side cost is the time between a source submitting data for transmission to the time at which the infrastructure invokes the underlying network 'send()' operation. Receive side costs represent the time between the end of the 'receive()' operation and the point at which the intermediate operator or the sink receives the data. Additional performance measurements comparing middleware performance to that of other high performance communication infrastructures appear in [5].

6 Conclusions & Future Work

In this paper we presented DSMI, a highly scalable and resource-aware approach to distributed stream management. The approach makes use of in-network data aggregation to distribute the processing and reduce the communication overhead involved in large-scale distributed data management. One of the important features of our infrastructure is its ability to efficiently and scalably deploy data-flows across the network. The run-time reconfiguration of the deployed flow graph in response to change in operating conditions and support for high-level language constructs to describe data-flows are other distinguishing features of the infrastructure. As a part of ongoing work we are examining how to avoid loss of updates and state in case of reconfiguration. We are also examining how to represent reconfiguration as a database-style transaction, motivated by similar work done by our group [25]. Another aspect of the infrastructure that is of particular interest to us is the closer integration of reconfiguration policy with the application level requirements. Overall, our architecture is a flexible, scalable platform for distributed processing of stream data.

Acknowledgements

The authors would like to thank the Emulab community for providing the infrastructure to conduct the experiments reported in this paper. The authors would also like to thank Delta Technologies [26] for providing data and a very useful use-case for the experiments.

References

[1] Samuel R. Madden and Michael J. Franklin. Fjording the Stream: An Architecture for Queries over Streaming Sensor Data. ICDE Conference, February, 2002, San Jose.

[2] Alexander S. Szalay and Jim Gray. Virtual Observatory: The World Wide Telescope (MS-TR-2001-77). General audience piece for Science Magazine, V.293 pp. 2037-2038. Sept 2001.

[3] A. Gavrilovska, K. Schwan, and V. Oleson. A Practical Approach for `Zero' Downtime in an Operational Information System. International Conference on Distributed Computing Systems (ICDCS-2002), July 2002, Austria.

[4] http://www.research.ibm.com/gryphon/

[5] G. Eisenhauer, F. Bustamente and K. Schwan. Event Services for High Performance Computing. Proceedings of High Performance Distributed Computing (HPDC-2000).

[6] V. Oleson, K. Schwan, G. Eisenhauer, B. Plale, C. Pu and D. Amin. Operation Information System – An Example from the Airline Industry. First Workshop on Industrial Experiences with Systems Software WEISS 2000, October, 2000.

[7] F. Bustamante, P. Widener, K. Schwan. Scalable Directory Services Using Proactivity. Proceedings of Supercomputing 2002, Baltimore, Maryland.

[8] E. Zegura, K. Calvert and S. Bhattacharjee. How to Model an Internetwork. Proceedings of IEEE Infocom '96, San Francisco, CA.

[9] http://www.isi.edu/nsnam/ns/

[10] S Babu, J Widom (2001) Continuous Queries over Data Streams. SIGMOD Record 30(3):109-120

[11] D Carney, U Cetintemel, M Cherniack, C Convey, S Lee, G Seidman, M Stonebraker, N Tatbul, S Zdonik. Monitoring Streams: A new class of data management applications. In proceesings of the twenty seventh International Conference on Very Large Databases, Hong Kong, August 2002.

[12] R. Koster, A. Black, J. Huang, J. Walpole, C. Pu. Infopipes for composing distributed information flows. Proceedings of the 2001 International Workshop on Multimedia Middleware. Ontario, Canada, 2001.

[13] Y. Ahmad, U. Çetintemel: Network-Aware Query Processing for Distributed Stream-Based Applications. Proceedings of the Very Large Databases Conference, VLDB 2004, Toronto, Canada.

[14] M. A. Shah, J. M. Hellerstein, S. Chandrasekaran, and M. J. Franklin. Flux: An adaptive partitioning operator for continuous query systems. Proceedings of the 19th International Conference on Data Engineering, ICDE 2003,.

[15] S. Madden, M. J. Franklin, J. Hellerstein, and W. Hong. TAG: A tiny aggregation service for ad-hoc sensor networks. Proceedings of the Symposium on Operating Systems Design and Implementation (OSDI '02), Massachusetts, Dec. 2002.

[16] M. J. Franklin, B. T. J´onsson, and D. Kossmann. Performance tradeoffs for client-server query processing. SIGMOD Record, 25(2):149–160, June 1996.

[17] D. Kossmann. The state of the art in distributed query processing. ACM Computing Surveys, 32(4):422–469, 2000.

[18] T. Abdelzaher, et al. ARMADA Middleware and Communication Services, Real-Time Systems Journal, vol. 16, pp. 127-53, May 99.

[19] Peter R. Pietzuch and Jean M. Bacon. Hermes: A Distributed Event-Based Middleware Architecture. In Proc. of the 1st Int. Workshop on Distributed Event-Based Systems (DEBS'02), pages 611-618, Vienna, Austria, July 2002.

[20] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In Proceedings of the ACM SIGCOMM '01 Conference. ACM Press, 2001.

[21] B. Y. Zhao, L. Huang, S. C. Rhea, J. Stribling, A. D. Joseph, and J. D. Kubiatowicz. Tapestry: A global-scale overlay for rapid service deployment. IEEE J-SAC, Jan 2004.

[22] Z. Cai, G. Eisenhauer, C. Poellabauer, K. Schwan, M. Wolf, IQ-Services: Resource-Aware Middleware for Heterogeneous Applications. 13th Heterogeneous Computing Workshop (HCW 2004), Santa Fe, NM, April 2004.

[23] http://www.emulab.net/

[24] http://dast.nlanr.net/Projects/Iperf/

[25] C. Isert, K. Schwan. ACDS: Adapting Computational Data Streams for High Performance. International Parallel and Distributed Processing Symposium (IPDPS), May 2000.

[26] http://www.deltadt.com/

[27] Shelley Q. Zhuang, Ben Y. Zhao, Anthony D. Joseph, Randy H. Katz, John Kubiatowicz. Bayeux: An Architecture for Scalable and Fault-tolerant Wide-Area Data Dissemination. Proceedings of ACM NOSSDAV 2001.

[28] R. O. Duda and P. E. Hart. Pattern Classication and Scene Analysis. John Wiley & Sons, 1973.

[29] W. Allcock, J. Bester, J. Bresnahan, A. Chervenak, I. Foster, C. Kesselman, S. Meder, V. Nefedova, D. Quesnel, S. Tuecke. Data Management and Transfer in High-Performance Computational Grid Environments. Parallel Computing, 28 (5). 749-771. 2002.

國立中山大學95學年度博士班招生考試試題

科目：**資訊管理論文評述(二)【資管系選考】**

2. 請仔細閱讀所附論文(Title: Factors affecting IT adoption within Indonesian SMEs: Manager's perspectives)，並且回答下列（50%）:

（1）評述這篇論文有哪些主要的缺點或問題？

（2）如果要進行一項類似的研究，你的研究議題、目標、相關文獻或理論基礎、研究方法、資料分析與討論等會如何設計或處理，以避免上述之缺點或問題？

Factors Affecting IT Adoption within Indonesian SMEs: Manager's Perspectives

Abstract

This paper presents some findings from an ongoing doctoral research project investigating the adoption of Information Technology (IT) within Indonesian small and medium enterprises (SMEs). The aim of this research is to compare the actual factors influencing Indonesian SME managers in IT adoption decisions with those factors suggested in the literature, which is drawn mainly from research in developed countries. An initial model of factors affecting managers' decisions regarding the adoption of IT was developed from the literature. This model includes external and internal factors and was used to develop semi-structured interviews with 35 managers of Indonesian SMEs in the area of furniture and handicraft. Factors influencing Indonesian SME managers in IT adoption decisions were extracted from the interviews and compared to the literature.

Keywords: Keyword Indonesian SMEs, IT Adoption, Manager, Factors

1. Introduction.

Small and Medium Enterprises (SMEs) form a significant proportion of the economy in many countries (Drew 2003; Ihlstrom et al. 2003). Managing SMEs is not, however, the same as managing larger companies since SMEs have fewer resources available and fewer personnel to manage the business (Fink 1998; Thong 2001). Accordingly, managing the information technology (IT) function in SMEs, including the adoption of IT, may be different from managing in large enterprises.

There is considerable literature explaining IT adoption within organisations, a portion of which is aimed at SMEs (for example Dutta and Evrard 1999; Ihlstrom et al. 2003; Scupola 2002; Walczuch et al. 2000). However, most of the literature is based on research conducted in developed countries, and so might not be able to explain completely the phenomenon of IT adoption in less developed countries such as Indonesia. Elliot (1996), Bagchi, et.al. (2004), and Gefen, et.al. (2005) argue that culture might influence the adoption of IT in different ways for developed and less developed countries.

This paper is a report of initial findings from an ongoing doctoral research project conducted by the first author. The research is investigating the adoption of IT by Indonesian SMEs. The aim of this paper is to compare the actual factors influencing managers of Indonesian SMEs in IT adoption decisions with the literature which is drawn mainly from research in developed countries, and to assess whether any differences found might be due to cultural influences.

In the next section, the theoretical background of IT adoption within SMEs is discussed. We then introduce our research method followed by a summary of the initial results. These

【背面還有試題】

are discussed and compared with the literature. In conclusion we highlight some factors where our findings differ from other studies.

2. Factors Affecting IT Adoption within SMEs

Rogers (1995) defines adoption as the decision to make full use of an innovation as the best alternative. IT adoption can be defined as the decision to use IT to support operations, management, and decision making in the business. (Fink 1998; Rogers 1995; Thong and Yap 1996).

The definition of SME itself varies among organisations and countries. As one alternative for a world wide definition, the World Bank use the following criteria for SMEs (IFG, 2002):

a) Small enterprise-up to 50 employees, total assets of up to $3 million and total sales of up to $3 million

b) Medium enterprise-up to 300 employees, total assets of up to $15 million, and total annual sales of up to $15 million.

In Malaysia, the Small and Medium Industries Development Corporation (SMIDEC) uses a combination of number of employees, invested capital, total assets, sales turnover, production capacity, and average income (SMIDEC 1998). In Indonesia, an SME is defined as any business organisation which possesses assets less than US$ 1 Million (excluding land and building) and has annual sales turnover less than US$ 5 Million (SMIDEC 1998). The organizations studied in our research would fit into any of the above definitions of SME.

The existing literature has documented some of factors that affect IT adoption by SMEs in developed countries (Chau 1995; Cragg and King 1993; Drew 2003; Dutta and Evrard 1999; Fink 1998; Ihlstrom et al. 2003; Thong and Yap 1996; Walczuch et al. 2000). Meanwhile Al-Gahtani (2003), Khalfdan et.al. (2004), and Tarafdar, et.al. (2005) focused on IT adoption in less developed countries but not specifically aimed at SMEs. So far only and Utomo and Dodgson (2001) focused on SMEs in less developed country, which is Indonesia.

The above studies identified factors that are divided into those internal to the SME and those external to the SME. These factors and the research that identified them are shown in table 1 and summarised in figure 1. These factors are mostly derived from research in developed countries, except for Thong and Yap (Singapore), and Utomo and Dodgson (Indonesia).

Internal		**External**	
Factors	References	Factors	References
Manager Perception toward IT such as benefit, cost, & risks # IT experience IT knowledge Perceived alignment with business strategy	(Cragg and King 1993; Fink 1998; Ihlstrom et al. 2003; Thong and Yap 1996; Utomo and Dodgson 2001)	***Government*** Public infrastructure availability Regulation # Subsidies # Training #	(Dutta and Evrard 1999; Utomo and Dodgson 2001)
Resources Financial resources availability Internal IT infrastructure availability Operational cost	(Drew 2003; Dutta and Evrard 1999; Fink 1998; Ihlstrom et al. 2003; Thong 2001)	***Customers*** Initiatives or pressure for IT adoption	(Drew 2003; Fink 1998)
Staff Staff readiness IT experience Staff acceptance for IT adoption [illegible]	(Chau 1995; Drew 2003; Fink 1998; Mirchandani and [illegible]	***Competitors*** Competitive pressure #	(Cragg and King 1993; Drew 2003; Fink 1998)

Table 1. Factors affecting SME manager's decision for IT adoption (from literature)

Factors not mentioned by our participants (see Table 2, section 4)

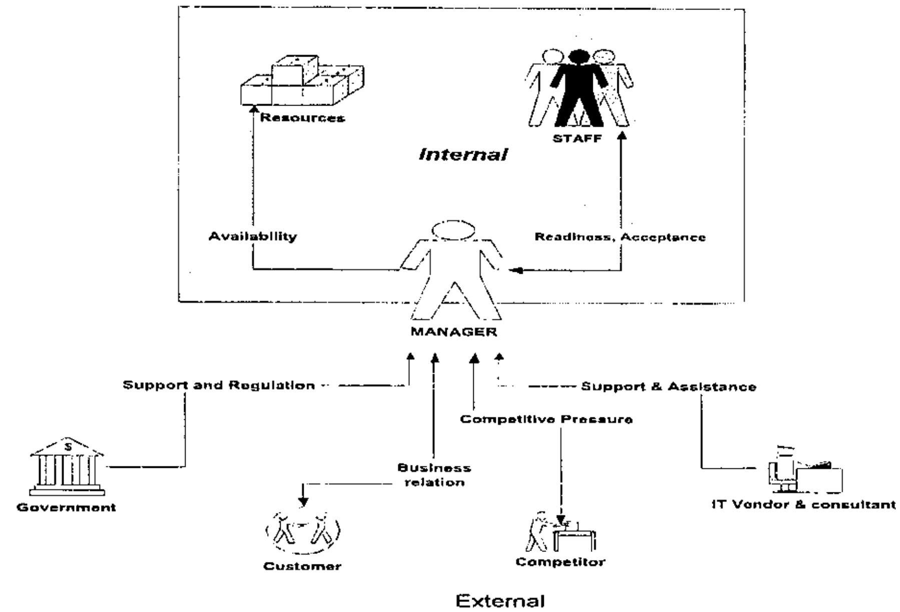

Figure 1. Factors influencing IT adoption decision making in SMEs (from literature)

In our research this table of factors was used as a guide when producing the questions for semi-structured interviews of managers of Indonesian SMEs. The relatively detailed questions allowed the interviewer to direct attention to these factors, while the open ended answers allowed interviewees to follow up these factors, dismiss them as insignificant, or suggest additional factors.

3. Research Method.

The above model illustrating factors influencing the SMEs' manager in IT adoption decision making was used to develop a semi-structured interview. Semi-structured interviews were selected to gain more in-depth and complete responses from participants, yet still provide a structure for further analysis (Creswell 2003; Leedy and Ormrod 2001; Williamson 2002). Semi-structured interviews also enable the interviewer to clarify participants' responses and explore a wider range of issues than might emerge otherwise.

The questionnaire was developed to guide data collection on the experience of the Indonesian SME managers during their IT adoption process, especially the factors that influenced them. The questions were asked to find out who were involved in the IT adoption process internally, the manager's perception toward IT, factors considered when the decision was made, and external parties involved. The questionnaire was originally developed in English and then translated into Indonesian by the first author who is native Indonesian speaker. The Indonesian language questionnaire version was tested on a few Indonesian students in Australia.

Once the final version of the semi-structured interview was ready, the field study was conducted by the first author between the months of December 2003 and February 2004. The participants were selected from a list of Indonesian SMEs in the furniture and handicraft industry and situated in Yogyakarta and Surakarta region in Central Java; the list was compiled from data provided by Indonesian Yellow Pages and Indonesian SMEs council and association. The furniture and handicrafts industries were chosen because they are significant local industry in which one of the authors had experience as IT consultant. As a manufacturing industry in an area of low labour cost, they are not obviously information intensive, but as most of their customers are international they need IT based communications. They also use IT in their internal business operations, but usually do not have a dedicated IT department.

All the participants already used basic computer applications for their business, such as office applications (for administrative functions, reporting, and book keeping), internet applications (for simple business intelligence, email, simple marketing and order tracking), and graphic manipulation applications for product design. Two of the companies had computerised inventory systems. The chosen (adjacent) regions of Central Java are considered as one the main centres of furniture and handicrafts in Indonesia; as all the SMEs are from the same region they face similar business environments (transportation, raw materials sources, export market, etc.).

From the list of more than 300 companies, 42 were unable to be contacted by any means. It is possible that the companies have already out of business or simply relocated without leaving contact details. Fourteen companies refused to be interviewed because either they did not use IT or the have stopped using IT. Further persuasion still did not change their mind. Thirty six companies accepted the invitations to participate and the rest of the companies on the list refused to participate with various reasons. In the end, only 35 companies can be included since the 36th companies were not fit into Indonesian SMEs definition. We decided to interview the 35 participants who have already had and used IT to find out what influences they were aware of.

The first author conducted all the interviews at the participants' premises or other places of their choice to enable participants to feel comfortable with the surroundings and therefore more relaxed during the interview. All the interviewees were the manager; 17 participants were also the owner of the SME, while the rest reported directly to the owner. The managers were selected because in SMEs almost all decisions are made by the manager and they are often directly involved in the IT design and implementation (Fink 1998; Thong and Yap 1996). We also found that in 18 SMEs with a dedicated manager (non-owner), the owner was heavily involved in decision making (including IT investment decision making).

All the interviews were recorded and transcribed by the first author. From the transcripts, a context analysis (Boyatzis 1998; Neuendorf 2002; Weber 1985) was conducted to extract the factors influencing the manager's decision for IT adoption. Phrases and sentences which were used similarly by several respondents were classified into one of the seven major categories and matched to factors from the literature (as shown in table 1) or added as new factors. From time to time the original recorded interviews were replayed and compared to transcripts to ensure the consistency of the data extracted. A text summary of participants' transcripts was made as a cross check on the factor mapping. From the factor table and participant's response summary, the factors that affecting the SME managers' decisions to adopt IT emerged. These findings are presented in the next section.

4. Initial Findings

Thirty five participants used various IT, ranging from office application to web catalogue. The participants' organisation has two main group of employee, the permanent administration staff (ranging from 1 to 50) and casual production staff (up to 700 at a time). The casual production staff were fluctuating as the order quantity change.

The factors extracted from the 35 interview transcripts were grouped into a table based on categories in the model drawn from literature above (table 2). Participants' response to the questions regarding the reason for IT adoption and factors they were considered for adoption were grouped and placed under the same heading as table 1.

Internal	External
Manager	***Government***
Attitude toward IT:	Public infrastructure availability
• Areas of IT usage	
• Future IT plan	***Customers***
• Initiatives for IT adoption	Initiatives or pressure for IT adoption
• Perceived impact of IT adoption	
IT experience and knowledge were not a problem	***Competitors***
Decision making style for IT adoption	Trend in IT adoption *
Resources	***IT Vendors & Consultant***
resources not a limiting factor	IT solutions availability *
	External consultant support
Staff	IT vendor support
Initiatives for IT adoption *	Initiatives for IT adoption *
Staff readiness	
IT experience	
Learning style *	

Table 2. Factors affecting adoption decision (extracted from interview transcripts)

* Factors not mentioned in table 1.

5. Discussion.

5.1. Manager

The study found differences between participants' responses and what the literature suggested as factors in management perception toward IT. In the literature management's perceptions, as they related to IT adoption, focussed on perceived benefit, perceived cost, and risk (Cragg and King 1993; Fink 1998; Ihlstrom et al. 2003; Thong and Yap 1996). The perceptions of our participants were mostly oriented toward future uses of IT rather than cost and risks. This attitude toward IT may come from their previous education or from information on the benefit of IT obtained from external sources (Chau 1995; Ihlstrom et al. 2003). They could see the areas where IT could help their business. They may have had a low perception of risk because they themselves often initiated the IT adoption, had a realistic expectation of the result, were personally involved in the project, and conceived future plans to adopt more IT innovations.

> *Before I built this company, I thought of what we needed in terms of facilities; a computer was one of them (participant 9)*
>
> *We intend to develop a bar coding system for our inventory next year (participant 1).*

All the decisions within participants' organisation were made solely by the manager with minimum or no input from the staff. This might be due to the centralistic style of management common in Asian countries (Elliot 1996) as well as being a common feature of SMEs (Rule and Attewell, 1989). This decision making style did not influence the manager in making IT adoption decisions but it might make the IT adoption easier, since all the decisions were made and enforced to the staff by the manager.

5.2. Resources

Within the participants' internal organisation, resources were not a problem. The participants were able to finance their IT adoption by allocating some of their capital, payment by instalments facility from the IT vendors, or donation from the participant's family members. This fact is somewhat different from the literature, where it is stated that SMEs would find lack of financial resources as one of the barriers to IT adoption (Chau 1995; Fink 1998; Ihlstrom et al. 2003). This difference is probably due to the fact that the prices of basic IT hardware in Indonesia are relatively inexpensive (there is also anecdotal evidence that software piracy is a big problem in Indonesia leading to availability of cheaper illegal software). In any case, hardware prices worldwide are lower than they would have been in some of the earlier studies. Other resources were also not a problem since the participants could acquire them if necessary.

> *When we need computers we only need to make a proposal to the owner (participant 7)*
>
> *We allocated some of our capital to purchase computers (participant 9).*

5.3. Staff

Within staff category, staff acceptance of IT adoption was also not the problem suggested by the literature (Fink 1998; Ihlstrom et al. 2003). Once the manager decided, the staff would usually comply or they would be moved to other sections which were not affected by the newly adopted IT. In other words, the manager imposed the adoption and assumed that the staff would support it unreservedly.

> *The sales force just accepted it (participant 3)*
>
> *If they don't like it, that is alright. I will assign them to other job that is not using computers (participant 4).*

Some participants even recalled that their staff were the ones who proposed the use of computers to solve their problems. This eventually led to the acceptance of the IT adoption.

> *One of my staff, who communicates with our overseas client frequently, asked me to buy computers and subscribe to the Internet (participant 15)*
>
> *We have computers, I think, in two of the departments and the rest of the staff indicated that they would probably work more effectively using computers, so actually it came from them as well (participant 1).*

Most of the participants' staff who operate computers were university graduates or required to be university graduates. As a result, computer literacy is not a major issue as suggested in the literature (Ihlstrom et al. 2003; Walczuch et al. 2000). After all, they only operate basic computer applications. This fact led to a self study learning style, where the staff would learn by themselves how to use the computer applications.

5.4. Government

The participants' only concern with the government was the lack of public infrastructure availability and quality. The public infrastructures of concern were telecommunications, especially access to the internet. Utomo and Dodgson (2001) suggested that this problem might have been solved by now, but the new government supported infrastructure project has not eventuated. In Europe, there appears to be more government support for IT infrastructure (Dutta & Evrard (1999).

> *We have difficulties in accessing the internet. It is slow and expensive due to the poor telephone cable (participant 9).*
>
> *There is no telephone/internet line here. We can't send our designs (participant 13).*

Mostly, the participants were not aware of any government regulations or subsidies that might have helped them in IT adoption. One the participants even believed, from the previous experience, that typical training and workshops provided by the government might not meet their requirements, since they were too general to be applicable. This confirms what Dutta & Evrard (1999) found in their study of European SMEs.

> *I am not aware of any computer training from the government (participant 14).*
>
> *The previous training is too much theory and inapplicable (participant 10).*

5.5. Customers

There are no differences between the findings from our field study and the literature with respect to customers. The findings confirm that the customers can initiate IT adoption by applying pressure to the participants (Drew 2003; Fink 1998). The major customers for all the participants were overseas customers who made up more than 90% of their annual sales and most of these customers prefer to do business by email.

5.6. Competitors

The competitors apparently had insignificant influence on the participants' IT adoption decision. Only one participant mentioned that he observed the trend of building websites for promotional purposes by other SMEs, yet he did not want to follow since his company's production capability might not be able to meet the increase in orders. Another participant said that he had evaluated a proposal for a company website, but had not adopted it and would not reconsider it in foreseeable future. This is in contrast to the literature's suggestion that competitive pressure could force the SMEs to adopt IT (Cragg and King 1993; Drew 2003; Fink 1998).

5.7. IT Vendors and Consultants

The IT vendors and consultants were perceived as necessary by the participants not only for their support but also as source of information on the availability of IT solutions that fitted their need, as suggested by the literature (Cragg and King 1993; Fink 1998). The participants believed that the IT vendors' and consultants' support would be critical during the deployment of the IT solutions due to the lack of technical knowledge within the participants' organisation. However, our study found that IT vendors and consultants also often gave the idea to the participants to adopt IT as a result of marketing efforts.

6. Conclusions and Future Works

From the comparison between results of the field study and literature, it was found that there were some similarities and some differences. The Indonesian managers agreed on the importance of public infrastructure availability, that pressure for adoption often came from customers, that external consultant and vendor support was important, and that success relied on staff readiness and IT experience (although in Indonesia this was almost always present).

The most apparent differences among the external factors concerned the government and competitors. The participants' only concern with the Indonesian Government was the lack of public infrastructure provided; there was little hope of government subsidies or training, and government regulations had no impact. While the pressure of competition seemed irrelevant to the IT adoption decision, managers did observe trends in IT use among their competitors. Among the internal factors internal resources, especially financial resources, were not a problem. This is different from what the literature suggested earlier (Fink 1998; Ihlstrom et al. 2003). Probably with the price of IT products becoming cheaper the acquisition of IT products was not a financial burden anymore. As for staff acceptance, it was found that this was almost guaranteed, probably due to the centralistic management style (Elliot 1996). Staff readiness again was not a problem since most staff were university graduates and had experience in using basic IT applications. Managers' general attitudes towards IT were not a factor in our study, since all were quite positive towards IT. In reality, their perceptions were more focussed on the benefits they could foresee from particular IT applications. The centralistic management style also proved favourable in IT adoption. There was no rejection from the staff once the manager decided, since they would be assigned to other functions if rejected the IT adoption.

As noted, these conclusions differ somewhat from previous studies, including that of Utomo and Dodgson (2001) which was also conducted in Indonesia. It may be that our semi structured interviews gave more scope for open ended replies, and therefore allowed the participants to introduce factors different from those included in the mainly questionnaire based models in the literature.

As we continue this project we will undertake further analysis of the interviews and of possible reasons for differences between our results and those previously found. We had originally intended take guidelines for IT adoption in SMEs such as those proposed by Fink (1998) and adapt them for use in developing countries such as Indonesia based on the results of our research. However, as it appears that the Indonesian SMEs studied are at least as successful in their IT adoption as those SMEs referenced in the literature, it may be more appropriate to use the Indonesian experience to produce improved suggestions for use in developed countries. This question will be studied further.

In general, our knowledge of IT adoption might be further improved by more studies (in both developed and developing countries) based on interviews rather than surveys, as many surveys have a tendency to prejudge what factors might be important.

References

Al-Gahtani, S.S. "Computer Technology Adoption in Saudi Arabia: Correlates of Perceived Innovation Attributes," *Information Technology for Development* (10:1) 2003, pp 57-69.

Bagchi, K., Hart, P., and Peterson, M.F. "National Culture and Information Technology Adoption," *Journal of Global Information Technology Management* (7:4) 2004, pp 29-46.

Boyatzis, R.E. *Transforming Qualitative Information: Thematic Analysis and Code Development* Sage Publications, Thousand Oaks, 1998.
Chau, P.Y.K. "Factor Used in The Selection of Packaged Software in Small Businesses: Views of Owners and Managers," *Information & Management* (29:2) 1995, pp 71-78.
Cragg, P.B., and King, M. "Small-firm Computing: Motivators and Inhibitors," *MIS Quarterly* (17:1), March 1993, pp 47-59.
Creswell, J.W. *Research Design: Qualitative, Quantitative, and Mixed Methods Approaches*, (2nd ed.) Sage Publications, Thousand Oaks, 2003.
Drew, S. "Strategic Uses of E-Commerce by SMEs in The East of England," *European Management Journal* (21:1) 2003, pp 79-88.
Dutta, S., and Evrard, P. "Information Technology and Organisation within European Small Enterprises," *European Management Journal* (17:3) 1999, pp 239-251.
Elliot, S.R. "Adoption and Implementation of IT: An Evaluation of The Applicability of Western Strategic Models to Chinese Firms," in: *Diffusion and Adoption of Information Technology*, K. Kautz and J. Pries-Heje (eds.), Chapman & Hall, London, 1996, pp. 3-31.
Fink, D. "Guidelines for The Successful Adoption of Information Technology in Small and Medium Enterprises," *International Journal of Information Management* (18:4) 1998, pp 243-253.
Gefen, D., Rose, G.M., Warkentin, M., and Pavlou, P.A. "Cultural Diversity and Trust in IT Adoption: A Comparison of Potential e-Voters in The USA and South Africa," *Journal of Global Information Technology Management* (13:1) 2005, pp 54-78.
Ihlstrom, C., Magnusson, M., Scupola, A., and KristiinaTuunainen, V. "SME Barriers to Electronic Commerce Adoption: Nothing Changes - Everything is New," in: *Managing IT in Government, Business & Communities*, G. Gingrich (ed.), IDEA Group Publishing, Hershey, 2003, pp. 147-163.
Khalfan, A.M., and Alshawaf, A. "Adoption and Implementation Problems of E-Banking: A Study of The Managerial Perspective of The Banking Industry in Oman," *Journal of Global Information Technology Management* (7:1) 2004, pp 47-64.
Leedy, P.D., and Ormrod, J.E. *Practical Research: Planning and Design*, (7th ed.) Merrill Prentice Hall, Upper Saddle River, N.J., 2001.
Mirchandani, D.A., and Motwani, J. "Understanding Small Business Electronic Commerce Adoption: An Empirical Analysis," *Journal of Computer Information Systems* (41:3) 2001, pp 70-73.
Neuendorf, K.A. *The Content Analysis Guidebook* Sage Publications, Thousand Oaks, 2002.
Rogers, E.M. *Diffusion of Innovations*, (4th ed.) Free Press, New York, 1995.
Scupola, A. "Adoption Issues of Business-to-Business Internet Commerce in European SMEs," Proceedings of The 35th Hawaii International Conference on Systems Sciences, Hawaii, 2002, pp. 1-10.
SMIDEC *Profile of SMEs in APEC Economies* Small and Medium Industries Development Corporation, Kuala Lumpur, 1998.
Tarafdar, M., and Vaidya, S.D. "Adoption and Implementation of IT in Developing Nations: Experiences from Two Public Sector Enterprises in India," *Journal of Cases on Information Technology* (7:1) 2005, pp 111-135.
Thong, J.Y.L. "Resource Constraints and Information Systems Implementation in Singaporean Small Businesses," *Omega The International Journal of Management Science* (29:2) 2001, pp 143-156.
Thong, J.Y.L., and Yap, C.S. "Information Technology Adoption by Small Business: An Empirical Study," in: *Diffusion and Adoption of Information Technology*, K. Kautz and J. Pries-Heje (eds.), Chapman & Hall, London, 1996, pp. 160-175.

Utomo, H., and Dodgson, M. "Contributing Factors to The Diffusion of IT Within Small and Medium-sized Firms in Indonesia.," *Journal of Global Information Technology Management* (4:2) 2001, pp 22-37.

Walczuch, R., Braven, G.V., and Lundgren, H. "Internet Adoption Barriers for Small Firms in The Netherlands," *European Management Journal* (18:5), October 2000, pp 561-572.

Weber, R.P. *Basic Content Analysis* Sage Publications, Newbury Park, 1985.

Williamson, K. *Research Methods for Students, Academic and Professionals: Information Management and Systems*, (2nd ed.) Centre for Information Studies, Wagga Wagga, 2002.

Winston, E.R., and Dologite, D.G. "Achieving IT Infusion: A Conceptual Model for Small Businesses," *Information Resources Management Journal* (12:1), Jan-Mar 1999, pp 26-38.