dpp

DEVELOPMENT POLICY AND PRACTICE

THE OPEN UNIVERSITY

NEW TECHNOLOGY AND THE INTERNATIONAL DIVISION OF LABOUR: A CASE STUDY OF THE INDIAN SOFTWARE INDUSTRY

BY

RICHARD HEEKS

DPP WORKING PAPER NO. 17

Development Policy and Practice Research Group

Faculty of Technology

The Open University

October 1989

NEW TECHNOLOGY AND THE INTERNATIONAL DIVISION OF LABOUR: A CASE STUDY OF THE INDIAN SOFTWARE INDUSTRY

BY

RICHARD HEEKS

DPP WORKING PAPER NO. 17

CONTENTS

New Technology and the International Division of Labour: A Case Study of the Indian Software Industry

Introduction and abstract

This paper presents a preliminary description and analysis of the current situation in an industry which is receiving increasing attention. The Indian software industry has been prioritised as an export earner for India and has expanded by taking up software production contracts and subcontracts from Western firms. ['The USA is by far the largest consumer of Indian software, with almost 90% of India's software exports going to that market' (CEI 1987, p3). This should be borne in mind when the terms 'First World' or 'Western' are used in this paper.]

The work has been subject to two forms of international division of labour. Firstly, in general, only the relatively less-skilled elements of software production have been undertaken by Indian firms. Secondly, export production has often taken the form of 'body shopping'. This is the process by which Western firms send a list of staff requirements to Indian software companies, who then send the required 'bodies' overseas to work for that client. In other words, the Indian workers have frequently been doing their work outside India.

New technologies, such as satellite communications and fourth generation programming languages (4GLs), are being increasingly employed in this industry, and these are having an impact on the division of labour. They seem to have relatively little impact on the skill division of labour, but satellites can reduce the extent of body shopping, and 4GLs are not increasing it.

The imperative to use these technologies derives mainly from external pressures. Western clients are demanding use of the latest software development techniques, and Indian companies cannot refuse. This process is aided by Indian government policy, which seems inclined to encourage new tools or techniques which will improve export earning potential.

It has been suggested in the literature on the international division of labour that automating technologies would lead to work processes which had come to be located within

Third World nations suffering a reverse location back on-shore to the First World. However, this view is now being questioned. It would appear from this study that only in the very long term might automation damage the opportunities for location of production in the Third World country.

The overall conclusion drawn is that these new technologies are currently increasing the opportunities for work to be done in the developing country, and are not contributing to a reverse location of production back to the First World.

1. International division of labour - literature and evidence

1.1 Early literature and ideas about the international division of labour

Recent discussions on the international division of labour (IDL) started from the situation that, as a generalisation, the First World had relatively capital-intensive production and made manufactured goods, while the Third World produced mainly raw materials, using labour-intensive methods. This was regarded as unsatisfactory by some development academics and by developing country policy makers seeking to promote industrialisation, because it locked the Third World out from benefits that the First World was perceived as having gained from industrialisation.

However, the possibility then arose for an internationalisation of production which would lead to the setting up of industrial facilities within the Third World. This was seen by some observers as the final stage in an '*international product cycle*', which would produce an internationalisation of production, when production would be freed from the constraints of having to locate near its main markets.

> "This cycle, it is argued, takes the following form. Manufacturing industry initially tends to be located close to the main markets for end-products. Then, as manufacturing technology improves and trade competition increases, production will appear in the next important markets. Ultimately the point is reached where manufacturing facilities become located according to where such factors as transportation and labour costs dictate." (Open University 1983a, p23)

Such an internationalisation and relocation of some elements of production from the First World to the Third World has occurred in reality, particularly in production of electronic goods and components, and in textiles and clothing.

From the days of Smith's and Babbage's analyses of how production was being achieved, and continuing through the 'scientific management' of Taylor and the production methods of Ford, there has been a strong concept that a job of work or production could be divided into smaller elements, and these elements given to different workers who would then become more specialised, more productive, and less costly than would have been the case if one worker performed all the different elements.

It soon became clear that such a division of labour was being applied to the internationalisation of production. Not all production was being transferred to the Third World, but only those elements of production which were most labour-intensive and, normally, least skilled. For example, in integrated circuit manufacture, Third World 'production' consisted largely of the final stages of assembly and testing, whereas design, mask generation, and etching were still done in the First World. One of the main reasons behind transfer of the more labour-intensive elements of production was that, as a generalisation, capital is relatively abundant and labour relatively scarce (and thus expensive) in the First World, while in the Third World the situation is reversed and labour is thus cheap.

Within the academic debate on the international division of labour, the focus on cheap labour highlighted an argument against such transfers of production, by Frank (1981) and others of the dependency school. They argued that once there was a 'maturing' of production in the Third World, labour would organise, would improve its bargaining power and would therefore be able to increase wages. As a consequence, the transnational corporations (TNCs) would simply relocate to another part of the Third World where labour costs were lower, leaving little, if anything, of developmental benefit for the original country. In other words, production would be lost with the erosion of the comparative advantage that the original nation had offered when its labour costs were comparatively much less than those of the First World. Evidence was cited in support of this theoretical idea by writers such as Rada (1980a, p126) who, in discussing the electronics industry, stated that 'off-shore installations are being moved from higher cost Asian countries ... to countries like Thailand and the Philippines in an effort to cut costs even further'.

The academic literature on new technology and automation was dominated by an equally pessimistic line of argument. As advanced by writers such as Rada, the pessimistic view also concentrated on the question of labour costs. It foresaw a 'comparative advantage reversal' as new micro-electronics-based technologies were used to automate production, thereby greatly reducing the labour intensity of production and thus the importance of

labour costs, and therefore once again making it favourable to locate production in the First World. 'The introduction of micro-electronics reduces the possibilities of establishing or locating manufacturing facilities in developing countries' (Rada 1980b, p88). 'This situation is producing a return of some manufacturing facilities to the most advanced countries and this will further concentrate industrial capabilities' (Rada 1980a, p143).

It is this view - that automation of production promotes a reverse location, back to the First World, of production that was set up in the Third World - that is one of the central points to be considered here.

1.2 Later views and evidence on the international division of labour

Before presenting details of the case study, it will be useful to take in some of the subsequent writings on IDL. Earlier ideas have now been subjected to a more sophisticated analysis which, while in some ways supporting what was said before, points out that there are complexities to the situation that have to be recognised.

Subsequent writings suggest that while labour costs are an important factor in relocation decisions, there are a number of other factors which also have a bearing. For one thing, labour costs must always be balanced against labour productivity - TNCs will see few benefits in paying someone one tenth of a Western worker's wages if they only produce one tenth the output in a given time.

There are also a number of other factors often associated with production in the Third World which will affect decisions about location of production. These include longer hours of work (extraction of more absolute surplus value); availability of certain types of labour such as female labour or labour with skills that are hard to obtain in the First World; greater labour compliance and flexibility so that restructuring and change should be easier than in the First World; control over unions; looser health and safety regulations; continuous shift working to allow 24 hour use of capital-intensive equipment; government concessions and inducements on tax, rates, etc.; political security and stability.

Similarly, factors have also been identified which would tend to reduce the desirability of moving out of a Third World location, even though labour costs there might have risen or been made less competitive because of automation. Companies value the ability to get easy access to the domestic markets of Third World nations, many of which were and are growing strongly in a number of sectors. It is also recognised that in almost any process, even one that

has been 'automated', there will be some labour component and thus some advantage to remaining in an area of lower labour costs. Finally, relocation is not a cost-free exercise, particularly because much production has become international, with elements in the final product coming from many different sources, as Ernst (1985, p343) describes:

> "Any change in the current pattern of manufacturing and sourcing would involve substantial costs, both in terms of closing down existing plants, re-shuffling supply and market networks, and in terms of benefits forgone that could be reaped from achieving even higher stages of internationalization."

Later evidence seems to support this. Elson (1988, p281), when writing about relocation to cheaper sites within the Third World, states that 'large electronics TNCs have sought out new and cheaper locations within South East Asia, though to date they have been relocating *expansions* of capacity in cheaper areas, rather than closing down existing plant completely'. Kaplinsky (1987, p16) argues that there may have been some relocation in the late 1970s and early 1980s, but that the factors noted above and an increasing degree of value-added locally 'are likely to make relocation within the Third World less likely'.

However, there are factors other than labour costs that may encourage a reverse location of production back to the First World. For example, the increased protectionism of Western nations, often via use of non-tariff barriers, can have an impact (van Themaat and Stevens 1987). Reverse location can also offer proximity to First World markets so that changes in demand can be responded to more quickly, and external economies from linkages to other First World industries and services.

Through a recognition of all these other factors, some authors have therefore questioned the over-emphasis on cheap labour which tends to exclude other variables. Wield and Rhodes (1988, p305) state that 'explanations of relocation based solely on cheap labour are misleading', and Elson (1988, p282) finds 'considerable evidence that the NICs [Newly Industrialised Countries] are moving beyond the stage where cheap unskilled labour working longer and harder than anywhere else is the main basis for their growth'. The importance of considering a wide range of factors is borne out by the example of the software industry in which considerations about skills and trust often outweigh those of labour costs.

The views of these writers on new technology and automation will be discussed later.

2. Software as a case study of internationalisation

Case studies cited to advance or refute the arguments on international division of labour have tended to be rather narrowly focused. Their scope covered the electronics and the textiles/clothing industry, and they concentrated on South-East Asia and, to a lesser extent, certain countries of Latin America. Although this paper chooses an example outside this scope, it is still focused on a single industry. The experiences within the Indian software industry, one of the largest in Asia, are in some ways similar to those of Singapore, Taiwan, and the like, but such a comparison ignores a large number of the poorer or less industrialised nations of the Third World. This point will be returned to in the conclusions.

In sectoral terms, previous discussion has, by and large, dealt only with manufacturing. The debate over whether software production represents a service or a manufacturing process is not a particularly useful one to engage in for the purposes of this paper. Software production is often regarded as being part of the service sector, yet both a production process and a product are readily identifiable. As Sauvant (1986, p82) notes 'the boundaries between data services and other goods and services are blurring, and the designation of an increasing number of products as "goods" or "data services" becomes a matter of choice'.

That software is being internationalised is fairly clear. Sauvant (1986, p86) describes the international product cycle that software has followed.

> "New software packages therefore tend to be developed and launched in the largest national markets for an application; the United States, given its extremely high penetration rate for computers, has acquired a substantial lead in this respect.
> Diffusion of software follows a product-cycle pattern: New software products are distributed into markets with lower levels of computer penetration - or less sophisticated users in a particular application - than the market for which they were originally developed."

However, he goes on to note that 'This transfer of technology to foreign markets does not generally require a transfer of actual production facilities, notably in the case of software packages; in effect, reproduction and transport costs for packages are very low, so that it is economical to duplicate the package itself at a central site'. Thus Sauvant claims that what is internationalised is the product rather than production.

In fact, certain types of production are being internationalised, but in order to understand this, it must be realised that there are different types of software products. The main division is between applications software - software programs as most people think of them, which do word processing, data analysis, graphics, and so on; and systems software -

encompassing software such as operating systems which run the computer, control the disks, directories, etc. Applications software ranges on a continuum from software packages, which are generally aimed at the mass market (e.g. WordStar, Lotus 1-2-3) and which are the products that Sauvant refers to; to specialist programs aimed at specific narrow markets and which hope to sell, say, a few dozen or a few hundred; to customised software, which is usually written as a 'one off' for a particular client.

One other type of software production is conversion work. This is the conversion of a program which runs on one computer (e.g. an IBM PC) to allow it to run on a different computer (e.g. an Apple Macintosh). Conversion work tends to be comparatively simple, and differs from normal software development in much the same way that translation differs from original writing.

Western companies' software packages are readily and widely available in the Third World (though often via piracy), but production of those packages has not been transferred to any significant degree. Certainly as far as India is concerned, companies in the First World have been happy to arrange distributors for their packages but have shown little interest in subcontracting package development work to Indian firms. Similarly, Indian companies attempting to write packages have generally achieved poor sales at home and abroad. The export earnings of the Indian software industry have therefore relied on other types of software, notably customised programs, written specifically to order for a single large customer, and some conversion work.

Therefore, while one may be justified in talking in general terms of the internationalisation of the software product, internationalisation of software production has so far remained restricted to certain relatively narrow areas.

3. Characteristics of the Indian software industry

Wield and Rhodes (1988, pp295-6) outline three main common characteristics of production processes that have been relocated to the Third World:

1. They are predominantly labour intensive.
2. They use 'unskilled' or 'semi-skilled' labour.
3. They are carried out by women workers.

Labour intensity and skills

Software production is a labour-intensive process, in which the human input has always been seen as central and in which labour costs have always represented the major cost element. However, unlike previous examples, software production is regarded as being a highly skilled process and, whatever the theoretical skill requirements, in practice it is largely closed to non-graduates, particularly in India, not least because of the necessity for an understanding of English by Indian software developers.

Having said this, it should be made clear that software production is not a homogeneous process, but is usually seen as being broken down into a series of steps, as shown below. In practice, actual software production is a lot messier than this simple picture would suggest, with various processes being conducted in parallel, and some iteration such that later processes feed back into improved repetition of earlier stages. Nevertheless, for the purposes of this discussion, it is more useful to keep in mind a fairly simple model.

Idea/Problem identified
|
Analysis and specification of requirements
|
Prototyping
|
Designing product
|
Coding (i.e. writing product)
|
Testing
|
[Delivered product]
|
Maintenance

This fragmentation of the software production process lays the groundwork for a division of labour, particularly because the earlier stages of analysis and design require higher levels of skill and experience, whereas those of coding and testing are relatively less skill-intensive but more labour-intensive. This split has tended to form the basis for the international division of labour in the software industry. The majority of software contracts awarded to Indian companies have allocated only the coding and testing stages to Indian workers. That is to say, Indian workers have mainly been used as programmers rather than as systems analysts or designers. [The overall term covering all or any of the jobs involved is 'software developer'.]

For example, a major retailer in the UK contracted out work for a series of programs which would take in details of financial transactions at branches throughout the country, integrate and then analyse them at their head office. The initial elements of analysing the problem, deciding on system requirements, and producing a detailed set of program specifications were carried out in Britain by a firm of UK consultants. Interaction with the Indian software company only began after this stage, when Indian programmers and software managers took up the program specifications and coded this into a series of software programs which, once tested separately and together, were installed in the UK.

Schware (1987, p1254) sees this as a general trend of collaborative software arrangements 'excluding the more difficult and creative activities in the software life cycle' from the Third World firms. He notes that 'This type of export arrangement may leave developing countries little room for creating self-reliant capacities in software production' because it will only develop lower-end skills.

The same is also true of software conversion work. Rada (1980b, p95) notes that 'a number of European-based software companies are transferring their conversion work under contract to countries such as India'. As in the previous example, 'it must be emphasised that the research and development which produce the essentials of the software are carried out in the developed countries, and this state of affairs will not change'.

These examples therefore set a fairly clear pattern of international division of labour, with the less skilled, more labour-intensive elements of software production being farmed out to Third World firms, while the more skilled aspects remain within the First World. However, within India at least, this characterisation of IDL is not the complete picture. It seems fairer to say that it represents the first stage of a process from which a different division of labour may develop.

The key to understanding this are issues of trust, risk and confidence. Even though Western clients carry out information gathering and site visits, they still perceive their potential collaborators in India as rather an unknown quantity, and they perceive risk in entrusting software projects to the Indian company. The perception of risk covers a number of elements, including the perception of local technological capacity and capability.

In order to reduce the risk, many clients choose to retain as much control as they can over production. The first contract they award to the Indian company will often be for a relatively unimportant task, and will usually come with tight specifications (i.e. all the

analysis and design) having been done by the client, so that there is little that can go wrong or be misinterpreted with the coding or code conversion. However, once the Indian company proves itself able to follow a set of instructions or specifications and to fulfill its delivery dates, then it may be entrusted with a little more of the software development process.

Thus, Indian software companies have tended to move up a 'trust curve', firstly taking on only the least skilled elements of software production, then also creating the design, and finally accepting responsibility for the entire software development starting from the client's 'statement of the problem'. The characterisation of only the relatively less skilled elements being contracted out is still true for most collaborations, but some Indian firms have moved on from just providing programmers to providing a 'complete software solution' to the overseas contractor.

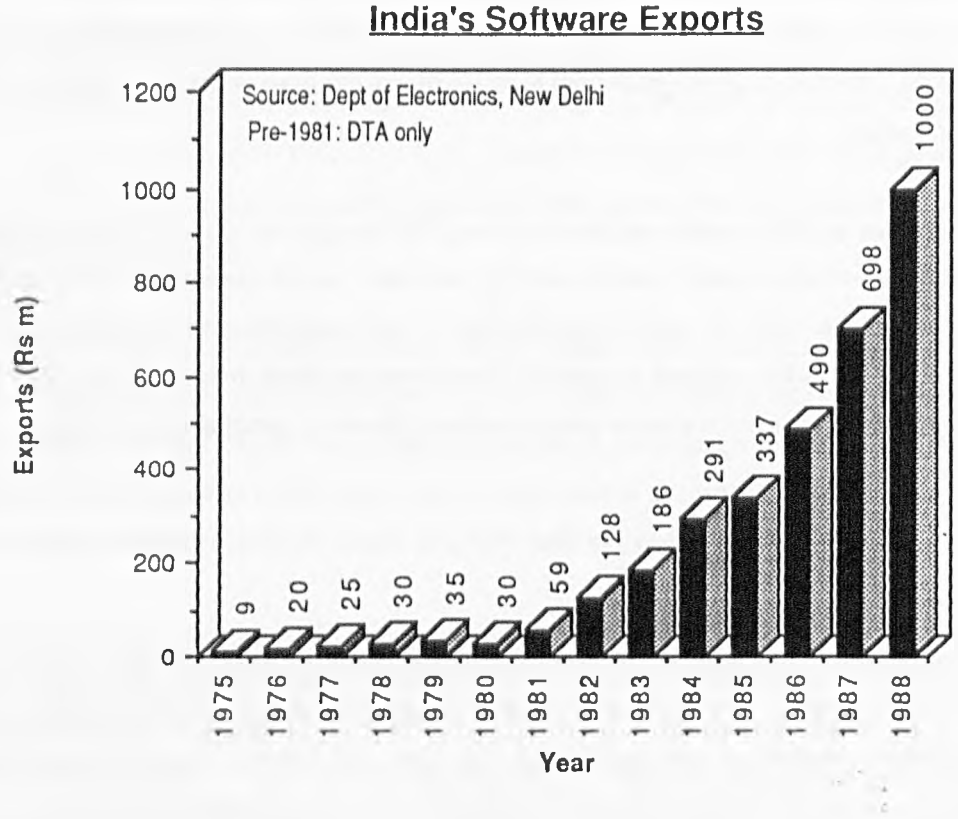
Of course, whether an Indian software firm moves up the 'trust curve' depends on the needs of the client, some of whom only wish to contract out tasks which are relatively unskilled. However, most clients wish to contract out the higher-skill tasks because the labour cost savings of hiring an Indian analyst are greater than those achieved from hiring an Indian programmer, and because such a move greatly reduces the managerial overhead on the client.

Gender

It is not the intention of this paper to focus on the gender issue. In previous discussions of IDL, gender was normally a central theme because a large proportion of the production work shipped offshore in textiles and microelectronics was done by women. There is no such trend yet to be seen within the software industry (though there is in the internationalisation of computer data entry services). Based on figures supplied by Indian software companies, one may estimate that between 5 and 10% of Indian programmers are female, and that women form an even smaller proportion of those sent abroad or those who become software managers. Clearly, this has its own implications in suggesting that women find it hard to progress beyond the relatively less skilled elements of software production. However it should also be noted that most of those interviewed, including women, seemed to feel that the software industry held relatively enlightened gender attitudes when compared to many sections of Indian industry and society.

Product intangibility

Software differs from previous IDL case studies because the product is an intangible one, being information-based. Software is therefore relatively cheap and easy to transport by conventional means but is also transportable by non-conventional means, i.e. via satellite. Its intangibility may also make it less susceptible to First World protectionist pressures or measures as more tangible products are.



Growth

Software is a booming sector. Global growth rates for computer software and services have been of the order of 15% per year during the 1980s (FT 1987, p1), which is far better than in most industries. Within the overall figures software sales are doing better than average (whereas data processing services are doing worse), and overall demand for software is projected to reach more than US $100 billion by 1990 (CEI 1987).

Growth rates in the Indian software industry appear even more spectacular, exceeding 40% per year in all but one of the past eight years, though from an admittedly small starting

point. [Note pre-1981 figures only include exports from the domestic tariff area, and not the export processing zones. EPZ exports prior to 1981 were, however, very small.]

3.1 Factors underlying the international division of labour

Trust and control

The desire of Western firms to retain control over the production process, at least until they build up more of a relationship with the Indian company, has been commented on above.

Labour costs

Of all the reasons for contracting work out to Indian firms, the most obvious is that of labour cost. For example, average monthly pay for an Indian programmer is around Rs3000 ($220), whereas for a programmer in the US it is over $2000. For a good analyst, the comparative figures might be $450 in India and $4000 in the US. In addition, Indian software developers appear to work rather longer hours than their Western counterparts, a typical working week being of 45 hours.

The apparently huge advantage that the Indian industry holds is complicated by two factors. One is that of non-salary costs. There are benefits to be paid in addition to salary (travel allowances, medical schemes, pension and provident funds), which can amount to 50% on top of basic pay, though of course these are found everywhere in the world to some degree. The cost of overheads is also high. Office costs in Bombay (still the centre of India's software industry, despite Bangalore's growth and promotion as India's Silicon Valley), are similar to those in Western capitals. The costs of software production tools and hardware, because of transportation and 65% import duty, are likely to be much higher.

Secondly, there is the problem of productivity, though without direct measurements it is hard to offer any definitive conclusions. The best Indian firms are definitely on a productivity par with competitors in the West, but it appears that other firms do not perform quite as well.

Both of these factors will serve to eat into the great difference that apparently obtains from looking at simple pay figures. Nevertheless, it will still be true that developing software in India is much cheaper than developing it in the West, largely because of low

labour costs. Estimates of the true comparative overall costs range from Indian software development costing roughly 20% of the US figure (Indian Engineering Export Promotion Council, quoted in Sauvant 1986, p87), to 30% (Kumar, 1987), to 50% (software manager for one of the large Indian software companies).

A potential problem noted in the literature was that of organised labour forcing wages to rise, which can encourage a relocation of production to cheaper cost countries. Wages in the Indian software industry do appear to be rising, and faster than those in other Indian industries, despite the lack of unionisation. A good estimate would be about 15% per year. However, this is broadly similar to the increases in software-related wages in many parts of the world, so Indian wages have remained roughly constant at one-tenth the level of US wages for a number of years.

Skills shortages

As well as the 'pull' of low labour costs there is another major factor, not much emphasised in other examples of IDL, which is the 'push' of skills shortages in the First World. Thomas (1987) notes that 'the number of programmers cannot be increased to match the growth in the number and power of computer systems.' The consequences of this are a huge software backlog of projects postponed or lengthened because of lack of personnel to work on them, and a driving up of salaries as too many jobs chase too few personnel.

Western firms have contracted work out to Third World software companies in order to help bridge the demand-supply gap. India's educated and technically proficient pool of software developers has appeared particularly attractive in this regard.

Other factors behind contracting out to Indian firms

The other factors mentioned in the literature that affect decisions about internationalisation of production - access to local markets, government incentives, and lack of unionisation - appear to have some importance for software.

The point was made earlier that companies in the First World may continue to undertake production arrangements in the Third World because they are seeking access to local markets. While India's domestic software market may have a large potential, it is small at the moment and an estimated 95% of software in India is pirated (IDC 1987, Vol II). Many of the leading US software companies have arranged local distributors for their products, but these arrangements have not generally included subcontracting of work. A

number of Western hardware firms, including ICL, Unisys, Hewlett Packard, and Xerox, have collaborative arrangements with Indian companies which may include some element of software development. This arises, at least in part, because government policy requires companies which import hardware to undertake a certain amount of exports. Only in the case of Unisys has production reached any significant level.

The government has created some concessions for foreign collaborations. For software development operations that are totally export-oriented, the regulation limiting foreign equity holdings to 40% has been relaxed, so that foreign companies can hold up to 100% of equity. Such wholly export-oriented firms are usually located in one of the six Export Processing Zones (EPZs) created by the Indian government. Of these, the main one of relevance to software is the Santa Cruz Electronics Export Processing Zone, located not far from Bombay's international airport. Firms setting up here are offered a number of tax, excise and export duty concessions, high depreciation allowances, and infrastructural subsidies. Such incentives will help to attract foreign companies to India, and the third and fifth largest software exporters in India are wholly foreign-owned. Despite these incentives, most software contracts undertaken by Indian firms do not involve any direct investment on the part of the Western client.

The weakening or prevention of unionisation has been seen as an important factor relating to decisions to relocate to the Third World. However, in the case of the software industry, there is little difference anywhere in the world because software is, in the main, non-unionised. In part, this is because there has never been a clear polarisation between 'workers' and 'managers' in software development. In part, too, the lack of unionisation arises from the superior weight of demand for software personnel over supply, which gives software workers a mobility and bargaining power that somewhat reduces the necessity for a union. Those who find conditions unsatisfactory in their company and who are unable to alter them by informal discussions with management tend to move on rather than attempting to organise for improvements.

Other factors such as compliance of labour, health and safety requirements, 24 hour working and political security seem to be of little relevance to the international software industry.

3.2 Factors underlying 'body shopping'

Factors favouring body shopping

It is possible for the work contracted to Indian companies to be carried out in India. However, a number of factors work against this:

i) (Lack of) trust and control. As stated already, Western companies perceive a risk in contracting out work to Indian companies (or, indeed, those of other Third World countries). They much prefer to have software development carried out under their noses, where they can tell at an early stage if there are problems and where they can see how these 'unknown quantities' actually work. Despite the movement of larger, maturer firms up the 'trust curve', there is a constant supply of new entrants to the Indian software export scene, and most of them begin with body shopping contracts.

ii) Hardware availability. Because of costs, export licensing in the West, red tape and linked export obligations in India, it has long been seen that one of the major constraints for the software industry in India is a lack of sufficient or sufficiently up-to-date hardware on which to develop software. Access within India to the latest hardware is improving, but one of the simplest ways in which this problem is currently circumvented is by getting the Indian software developers to come to the West where they can work directly on their client's machine.

iii) Interaction with clients. The development of software requires continuous interaction between client and developer, to discuss problems, misunderstandings, progress, and so forth. In the absence of adequate communications technology, interaction takes place best face-to-face and this will mean the developers going to the client, rather than vice versa. In such circumstances, it is usually easier for the developers to stay with the client all the time rather than keep hopping back and forth, particularly when the developers have not been responsible for the analysis and design.

iv) Skills shortage and poaching. Some firms in the US and Australia, where work and residence permits are relatively easy to come by, see Indian labour as a way of solving their personnel shortages on a longer-term basis. Foreign firms that have attempted to recruit Indian staff directly by advertising in Indian papers have been met with strong protests about so blatant a drain of skills. These firms therefore hire Indian contract workers with the specific intention of poaching them, so that they do not return to India. The Indian software developers are normally only too happy to fall in with this. Indeed, many of

them see software as a career with the goal of obtaining a US resident's 'green card'. Estimates vary as to the extent of staff losses. Indian companies will publicly admit to losing between 5 and 35 per cent of staff sent abroad per year, but the actual figure may well be higher.

v) Flexibility. Western companies can 'order', say, 50 COBOL programmers for a year, with no worries that the programmers will be demanding jobs at the end of their contracts, with no fears of any legal redress if they are sent back to India, but with the knowledge that, if it proves useful, many can be taken onto the payroll permanently at the end of the year, with little chance that their Indian employer can do much about it. Such contracts are mainly for programmers, though firms moving up the 'trust curve' are starting to send out systems analysts and project managers.

Indian software developers working on contracts from Western clients have therefore tended to undertake their work not in India, but at the client's site in the West, thereby setting up an international flow of labour back and forth between India and the West, though with a greater flow out than back. This flow, and the system of contracts that it entails, has come to be known as 'body shopping'. This system is not without its problems and criticisms.

Factors working against body shopping

Body shopping has come in for moral criticism from some commentators who, picking up Rajiv Gandhi's comment that he sought to take India forward into the 21st century, have described body shopping as '21st century slavery'. The government itself has criticised body shopping as 'leading to poor value addition and outgo of foreign exchange' (BusinessIndia 1988, p14). Thirdly, such methods of contracting are a major drain on the skills of India, not just in the short term by redirecting talent from work related to India's development to work that helps the West, but also in the longer term through permanent loss of those workers who do not return. This is felt at both the national level and by individual firms. It costs thousands of dollars to educate and train the top-level specialists who go into the best software jobs, yet all this can so easily be lost.

Some have seen body shopping as a positive thing for India by allowing those who go on such contracts to gain skills and experience that they would not otherwise have. A problem here is that Western clients are normally unwilling to accept inexperienced Indian staff and train them. However, Indians working abroad do gain experience and it was hoped that knowledge built up on such export contracts could then be transferred to benefit the domestic

software market. However, this hope appears flawed. Firstly, staff who gain any useful skills may decide they prefer life in Santa Barbara to that in Bombay, and fail to return.

Secondly, in general, domestic market work is seen as the 'poor cousin' of overseas work, and software developers often cut their teeth on domestic projects and then graduate onto work abroad. Thus, the flow of experience and staff tends to be from the domestic to the export markets and not vice versa. Only where software developers have been persuaded to undertake export work within India, have they been more willing and able to transfer any new skills gained onto domestic work, either by transferring back to such work themselves, or by training their domestic-oriented colleagues.

When work is carried out overseas, costs are higher than for offshore export work. Programmers' salaries are still paid into their bank accounts but additional costs are incurred for visas, transportation, and accommodation. A generous living allowance is also paid, and some software developers are accompanied to the West by their families. Lastly, the largest Indian firms need to cover their overheads for maintaining permanent offices in the West. The scale of these additional costs is such that, on body shopping contracts, Indian companies are only able to undercut US competitors by between 5% and 25%, compared to figures of 50-80% for offshore development work.

Finally, there are also signs that protectionism is becoming an issue. Ed Yourdon, one of the top consultants in the US has been warning that the US software industry 'will go the way of the Detroit auto industry, decimated by cheaper foreign imports' (Foremski 1988, p18). On a more proactive note, Lamb (1988, p25) states

> "The prospect of off-shore programming and packages produced in developing countries has not pleased everyone.
> The Data Processing Management Association (DPMA), the US organisation, has threatened to call for import restrictions on software from countries engaging in unfair competition. 'We are feeling the pinch,' says a DPMA spokesman."

The only concrete result so far of these pressures has been a tightening up of visa regulations for Indian software developers going to work in the US. Whereas previously almost all visas were granted, now some 40% are rejected according to US consular officials. This has made body shopping more difficult to undertake.

Despite the problems of body shopping and the greater cost advantages of allowing export work to be done offshore, Western companies still generally perceive the disadvantages of offshore development as outweighing the benefits. In these circumstances, low labour costs

become secondary to the problems of trust, hardware availability and interaction. However, Indian firms are trying to reduce the amount of body shopping in which they are involved, and in this they appear to be having some success. This is particularly so for those who are able to build up a longer-term relationship with their US clients, which will help to offset some of the trust/control problems. One of the largest software firms stated that while 'some years ago', less than 20% of their work was carried out in India, this had now risen to 'just under 50%'.

There are therefore factors working both for and against body shopping. New technologies may also affect it, as will be discussed below.

Summary

To summarise the specific features of this case study:

1. Location in India which, although unique, offers some commonality with other countries in the NICs or nearly NICs category.
2. Appears to transcend or confound traditional ideas about divisions between manufacturing and service sectors
3. Requirement for a high level of skills
4. Gender aspects apparently dissimilar to those of previous examples
5. An information-based, not a material, product
6. A rapidly growing sector
7. Little direct investment by foreign firms
8. International mobility of labour
9. Use of a non-standard set of new technologies (see next section)

In concluding this initial section, it can be said that internationalisation of software production has occurred later than that of textiles and electronics and is different in a number of ways to what has gone before. A study of the Indian software industry can, therefore, provide us with some new evidence, which can provide new insights into the international division of labour.

Although software development is a skilled profession, the IDL associated with work in the Indian software industry was, until recently, largely characterised by relatively less skilled elements being contracted out to Indian firms. These firms do, however, move up a 'trust curve', which leads to their taking on more of the software lifecycle. The IDL is

encouraged by low labour costs, shortage of personnel in the West and, in a few cases, by access to local markets and government incentives. The IDL is also characterised by a substantial amount of contracted work actually being done in the First World by Indian workers who are flown over there, in part because the problems of trust, interaction and staff poaching can be as important, if not more so, than low labour costs. However, problems with 'body shopping' are pressurising some firms into changing this method of working.

4. The two technologies - 4GLs and satellites

The new automated technologies considered within previous examples of IDL have largely been concerned with the application of microelectronics to the capital goods which created the product. In this sense, the new technologies being applied to software are rather different. One (satellites) is concerned with transportation of either the product or intermediate inputs in the production process, while the other (4GLs) is not microelectronics-based at all. A further difference is that software-oriented new technologies, such as 4GLs and expert systems, are automating mental work more than manual work.

Ideas about these two new technologies must also be kept in proportion to their use. While this is increasing, the majority of software development work in India still uses neither. This is partly due to reasons of costs, though these are decreasing. The costs of both technologies - purchase and use of 4GLs, payment for data transmission on satellite - may put them out of the reach of individuals and the smallest firms, but they are affordable by many companies in the Third World (though 4GLs much more so than satellite links).

Nevertheless, these technologies do automate parts of the software production process, and study of their use helps to indicate trends and can inform the debate on technology and the international division of labour.

4.1 Satellites

Rada (1983) compares the advances in telecommunications, of which satellite transmission is a major part, as being akin to the effect of the steamboat and the railways in the last century, in allowing an internationalisation of trade in 'intangibles' and thereby allowing comparative advantages in the production of services to become manifest.

What satellites are able to offer to software development and related work is a fast, automated and accurate method for transportation (i.e. transmission) of data, and a method which can save money provided the volume of data transmitted is high enough. Satellite links can transport finished products but also intermediate ones during the production cycle. While other communications methods, such as telex, could sometimes be used, they are not so quick, reliable or confidential.

There are only two well-known cases of the use of satellites in the Indian software industry. In one case, a satellite was used to link a large US insurance corporation office with software companies in the UK and India. This enabled a mainframe in the US to be accessed each day for 8 hours by software developers in the UK, then for 8 hours by workers in San Francisco, and finally for 8 hours by users in India.

The other case is of Texas Instruments (TI), which acted as the pioneer for setting up a software company in India using satellite communications. It began operations in 1987, using satellite communications to link an office in Bangalore to others in the US and UK. The Bangalore centre works mainly on the development of software for semiconductor and integrated circuit design, and it has so far exported more than $4m worth of software.

The major use and advantage of satellite links is the ability of the Indian user to access hardware abroad that is not available within India. Using a satellite in this way means that more software development work can be done offshore, in India, with fewer workers having to travel to the West. It can also offer significant savings if the only alternative is purchase of the hardware and shipment to India.

As well as affording access to the right hardware without the need to ship it out to India, satellite communications also enhance the ability of Western client and Indian developer to interact on a daily basis. Lamb (1988, p24) states that 'Telex, telephone and satellite links keep coders in Bombay abreast of developments at the clients' premises'. What the satellite link offers specifically is the ability to transmit programs under test back and forth. One TI manager is quoted by Raghavan (1988, p14):

> "The satellite link is indispensable....Specifications are constantly evolving and we have to keep upgrading the software. So, interaction with software and IC designers elsewhere is vital."

TI's clients are able to try out the software as it is completed, or in prototype form, make comments, have their own staff add subroutines, check it against original or updated specifications, and so on, several times during the overall development process. Outside

body shopping contracting, full interaction normally takes place only once or twice during software development and involves a group of managers and designers flying out from India to the client's premises with the most recently developed parts of the program and then flying back once modifications have been discussed. Such a paucity of interaction fits in poorly with the iterative nature of software development. The problem becomes even more acute with maintenance of software once it has been delivered. Indian companies either have to leave a programmer or two on site with the client, or clients themselves have to take over maintenance responsibility. Satellite transmission overcomes these problems.

As noted previously, lack of interaction is perceived as a major problem by potential Western clients and helps to reduce the level of trade in which the Indian industry could become involved. With greater and improved interaction, not only can more software development work remain in India, but the reduced risk for Western clients may encourage moves up the 'trust curve' to allow more of the software lifecycle to be undertaken by the Indian software companies.

This encouragement offers some hope of improving the skill level of tasks contracted to Indian companies but, as yet, the impact of satellite links on the skill division of labour has not been great. In the insurance company example, only the less skilled elements of work were contracted out to the Indian company. The same is true at TI. It is still exceptional that a project should be conceived in Bangalore. Far more often, the requirement is stated by one of TI's offices abroad, and the program specifications are also developed in the West, so the Indian workers only undertake the relatively less-skilled tasks of coding and testing.

However, the use of satellite communications does attack two of the major problems associated with body shopping - the problems of hardware availability in India, and the need for good client-developer interaction - and so allows export workers to remain in India. By reducing the need for body shopping, satellite links, though expensive, can help to save on high costs associated with body shopping for travel, visas, living allowances, etc.

The Texas Instruments office would not have come to India but for the satellite link and it now employs over 100 people in its Bangalore office - people who, for the most part, stay in India to undertake their work. Plans are also in progress to set up a 'software technology park' linking a cluster of software firms in Pune via satellite to firms based in Boston, USA. Thus it would appear that this new technology is increasing the opportunities for work to be done in India and is not encouraging a reverse location of production back to the First World. But then, this is not too surprising - a technology which operates as a kind of transportation automator is always likely to facilitate the internationalisation of work. To reiterate

Rada's point above, satellites are not harming, reducing or reversing India's comparative advantage. Instead they are allowing it to be more 'manifest' and accessible.

4.2 Fourth generation programming languages (4GLs)

For those unfamiliar with programming, it can be difficult to conceptualise 4GLs. An analogy that can be useful is that of painting, and to think of creating the software program as being analogous to painting a huge mural. Computer programs are written using 'words' and commands from a programming language. These languages have developed over the years from the 1st generation (machine code) to the 2nd generation (assembler languages) to 3rd generation (BASIC, COBOL, etc) to the 4th (Oracle, Focus, Linc, etc).

Using the painting analogy, this would be akin to moving from a small camelhair brush to an ordinary quarter inch brush, to a larger brush, to a small roller. Just as one stroke of a roller puts down a lot more paint than a brush, so a command line written in a 4GL is more powerful and causes a lot more computer operations than a line written in a language of an earlier generation. At each stage, you lose some ability to do the detail but, considering the immensity of the task to be done, you get the overall job done much more quickly and you have more time left to think about design.

4GLs and body shopping

4GLs can have an impact on the factors that influence body shopping.

Certain 4GLs can affect the problem of hardware availability because they are portable across a wide range of hardware environments. In a number of cases, Indian software firms have been able to complete the bulk of software development work in India on one of their own microcomputers and then simply transfer this to the Western client's minicomputer or mainframe. This will tend to increase the opportunities for work to be carried out offshore rather than on-site.

Lamb (1988, p24) claims that 'The emergence of software methodologies and software engineering tools adds to the confidence users can have in long-distance development'. 4GLs fit into this issue of trust and confidence by their applicability to the technique of prototyping. A prototype software system looks just like the finished application on the screen but lacks some of the essential underlying machinery.

Prototyping is used in the design phase when the prototype is presented to the client, who then gets a much quicker and better understanding of the workings of the new system than would be possible on paper. More valuably, the communication between client and developer is thereby enhanced and the system can be modified to fit the client's needs more accurately. This technology has an obvious utility in India for export contracts (particularly if coupled with satellite transmission), where misunderstandings between clients and developers from different nations and cultures can and do arise. Giving clients the confidence that they could see and comment upon a prototype system should lead to a decreased perception of risk in contracting work out to Indian firms, and a possibly decreased need for body shopping.

However, while it might improve the client's confidence to allow work to be done offshore, in reality prototyping requires a much more interactive style of software development with more frequent discussions between client and developer than traditional methods. In the absence of widespread availability of satellite communications, most of those interviewed who used prototyping undertook it at the client's site.

Not all 4GL projects require prototyping, and prototyping does not necessarily require a 4GL, but there are some further reasons which limit the chances of 4GL projects being brought offshore. 4GLs are associated with a shortening of the coding and testing phases of development. Because of this, a number of clients had felt that the additional organisational and communication problems of carrying out work offshore were no longer compensated by the simultaneous reduction in costs, and they had therefore decided to keep work on-site.

Finally, since 4GL skills have a higher premium placed on them in the West than 3GL skills, Western firms seeking to take on Asian software developers permanently are more than happy to poach those with 4GL experience.

Because of all these factors, some firms reported an increased likelihood of 4GL projects being brought offshore, a few more reported that 4GL projects were more frequently carried out at the client's site, but the majority felt that no consistent trend was yet visible. Therefore, at the moment, one may conclude that 4GLs have not been associated with either an increase or a decrease in the level of offshore, as opposed to on-site, software development. Their overall impact on body shopping is therefore not yet certain.

4GLs and skill division of labour

The apparent impact of 4GLs on the international skill division of labour was similar to that of satellites. 4GLs are likely to be one element encouraging movement up the trust curve, because Western clients feel more confident about Indian companies which have abilities to use up-to-date technology. Some contracts involving 4GLs were of 'coding and testing only', but more were 'turnkey' in which the Indian company was responsible for the whole software development process.

Because so many factors are involved, it is hard to disentangle the extent to which outcomes arise because of use of 4GLs or because of company size, reputation, relationship to client, type of contract and so on. However, there are some reasons to believe that 4GLs are associated with a greater integration of previously separate software production steps, apart from the trust factor already mentioned. 4GLs can considerably reduce the length of time required to undertake the coding and testing stages. Because of this, it is sometimes pointless to subcontract only those elements, rather than the whole production process. More importantly, when prototyping is used, software development becomes much more iterative and integrated - it becomes much harder to separate out the lower skill elements from the higher skill ones. In this case, the same team was frequently seen to undertake parts of the analysis and all the design, coding and testing.

Labour costs and comparative advantage reversal

Estimates vary greatly as to the increase in productivity that 4GLs can offer over use of a 3GL, but typical claims are those of Schofield (1987) who instances a 4GL giving an expected 80% saving on programming time and 95% saving on maintenance costs.

Taken simply, it will be seen that, by reducing the amount of labour input required to the software development process, 4GLs reduce the labour-intensity of that process and therefore reduce the comparative advantage that nations like India have to offer because of their much lower labour costs. In some ways, 4GLs may thus be seen as similar to production automation technologies discussed in other IDL examples.

A brief calculation will serve to highlight this. There are no definitive figures on software costs, and what figures there are tend to vary quite widely. Therefore, the calculation should be viewed merely as a very broad indicator of trends - it is not intended to be accurate.

Kumar (1987) estimates that normal software development costs split 9:1 between labour and other costs in the West. One may then compare this with a situation in which the labour element falls to 75% of total cost through the use of more productive techniques, such as 4GLs. These figures can be used to calculate the relative competitivity of Indian software costs compared to those in the US, based on Kumar's assertion that labour costs in India are one-tenth of those in the US, but that other costs (e.g. software production tools and hardware) are twice as expensive. In this imaginary example, dollars are used as the units.

	90% labour cost		75% labour cost	
	US	India	US	India
Labour costs	$90	$9	$75	$7.5
Other costs	$10	$20	$25	$50
Total	$100	$29	$100	$57.5

In this calculation, a move from more labour-intensive production (90% labour) to the less labour-intensive situation (75% labour) roughly halves the international competitiveness of the Indian industry. Of course, this example is likely to be inaccurate, and is very simplistic, taking no account of other factors such as overall hours worked, productivity, and qualitative aspects. However, whatever the actual figures, the calculation does point to a worsening of India's competitive position when trying to compete using more automated, less labour-intensive methods.

The other point to note, though, is perhaps even more important. This is that although the comparative advantage is reduced, it is by no means reversed. As long as software production requires a fair degree of labour input, areas of the world which can offer lower labour costs will still retain some cost advantage, albeit a steadily reducing one should automation increase. Indian firms certainly found themselves still undercutting US companies on 4GL contracts.

Any comparative advantage reduction will be counterbalanced to some extent by the wage levels associated with 4GLs. In the West, wages are rising for 4GL specialists, because demand is much greater than supply. For example, Computing (1987) reports that 'ICL's Quickbuild fourth generation language is proving a happy hunting ground for programmers and analysts who can command salaries of £35,000 a year after only six months' experience'. Such salary figures are more than double what an equivalent 3GL software developer would get. By contrast, the major firms in India all stated that there were no pay differentials

between those staff using 4GLs and those using other languages. This is mainly because staff tend to be generalists rather than 4GL specialists. In other words, 4GL experience is only one skill among a range, so much less of a premium is put on it. Such a difference in pay levels, because of severe 4GL skill shortages in the West, will go some way to cancelling out the reduced advantage that less labour-intensive production creates.

One may make a tentative conclusion that, while the spread of 4GLs may impair India's comparative cost advantage to some extent, this advantage is unlikely to disappear until software development becomes considerably more automated and even then it may remain. Hence, if automation technologies discussed in previous IDL examples truly allowed a reversal of the comparative advantage of Third World nations, 4GLs are not directly comparable because they only reduce, rather than reverse, the advantage.

4GLs and personnel shortages in the First World

It was noted above that one important factor operating in the internationalisation of software production was the shortage of skilled software personnel in the West. There have been fears that use of automated tools like 4GLs could overcome this shortage, thereby reducing the opportunities for software exports from the Third World. However, it would appear that the personnel shortage in the West is sufficiently large that 4GLs and related technologies will take some time to have an impact on it.

A recent US Department of Defense study concluded:

> "The demand for software is increasing at an annual rate of 12%, but the number of expert software professionals is growing at only 4% annually. Present software productivity tools boost the resource growth rate by only 4%, leaving a 4% annual shortfall that will amount to a shortage of as many as one million software developers by 1990." Schware (1987)

There is no clear agreement as to when 4GLs and the like will begin to bite effectively into the demand-supply gap. Schware himself predicts that 'Not until about 1990 will these technologies and methods be widely enough used to decrease demand for software labor'. Most others see this as an overestimate and agree with the DoD study that the gap will be around for some time to come. One senior software consultant in the UK, quoted by Lamb (1988, p24) states, 'I cannot see the demand for information technology development being met from UK resources over the next five to ten years'. Therefore, even laying aside all arguments about low labour costs, it would appear that demand based on the skills shortage will continue in the medium term despite the use of new productivity tools like 4GLs.

Why 4GLs are used by Indian companies

The impact of 4GLs on comparative advantage and on skills shortages in the First World appears to present dangers to the internationalisation of software production only in the long- rather than medium-term. Even so, there are problems involved with using 4GLs, such as those of cost, training, return on investment and so on. Despite these, more and more Indian firms are using 4GLs in their export work. The increased use of 4GLs seems to derive in large part from the drive in the West to use these new software technologies. If Indian companies wish to remain integrated with the global market, they will have to follow the trend.

The bulk of India's software exports is made up from contracts involving custom-written software for a single client. Overall, professional computer services, of which custom-written software forms a large part, is one of the most strongly growing sectors in the US, predicted to grow by 20% per annum between 1985 and 1990. 'One reason for this is a new determination on the part of commercial companies to have software written especially for them, to give them a particular competitive advantage' (CEI 1987). This determination is also linked to use of 4GLs:

> "Whereas three years ago, the main reason for buying a 4GL was to attack the growing list of applications to be developed over the year, now companies want the facility to produce systems quickly in response to changes in their business direction." Sweet (1988)

As an example of this increased use, it is now estimated that 80% of computer installations in the UK have some kind of 4GL product in use (Cornes 1987), and the US figure is probably higher.

4GLs are therefore becoming the 'weapon of choice' when writing software in the West and this increasing use obviously has implications for Indian software firms. One Bangalore-based software firm recently found problems winning contracts in the US. The only way to successfully secure a contract was to include use of a 4GL, with its consequent reduction in overall cost estimates. Many other firms have found it necessary to adopt a very pro-4GL attitude with Western clients, and stated that they 'always bring up 4GLs as an option for any new client'. Of those firms surveyed in 1989, over two-thirds had used a 4GL on at least one export contract, compared to less than half in 1988.

One may conclude that software firms in India which are oriented to the export market will find themselves having to use 4GLs on some contracts if they wish to continue to compete. Certainly, India does not have a monopoly position in Asia in offering low-cost, high-

quality software labour, and if Western companies wishing to use 4GLs do not find a market for their contracts in India, countries in South East Asia will no doubt take them up. This is not to say that the advent of this new technology *per se* is increasing the chances for internationalisation of production, though the particular shortage of 4GL skills in the West may help this. What it does say is that if this new technology is ignored by India, then opportunities for software exports will be reduced.

In conclusion, one may say that only in the longer term might the currently perceived impacts associated with 4GLs have a negative effect on the chances of software work being brought to India. At present, an ability to work with 4GLs is helping Indian firms to win more export contracts. Therefore, this automation does not currently appear to be leading to a reverse location of production back to the First World. However, such conclusions may relate to the specific factors governing software, in which shortage of skills may be as important as labour costs.

4.3 Indian government policy

The main instrument of the Indian government's software policy was the 'Software Policy on Computer Software Exports, Software Development and Training', released late in 1986 by the Department of Electronics (DoE). It is generally credited with having created the identity of an Indian software industry.

It was also the intention that the policy should have 'liberalised the import of systems of contemporary technology for the purpose of development of software for export projects' (Govt of India 1987, p7). In the case of the two technologies covered here, it seems to have had some success, since access to the technologies was next to impossible prior to 1986 but is now relatively easy. Policy changes have occurred because policy makers realised that hampering access to new technology would only damage the export industry.

Of Texas Instruments' satellite connection, Poe (1987, p96.10) says 'Although India's Department of Electronics took the lead, several other government agencies were involved, and as many as 27 regulations had to be changed to permit the new arrangement. It was more than two years before TI could send its first transmission via the satellite, and there is still more red tape waiting to be cut.' However, TI's groundwork will have helped to shorten the application time for subsequent companies and some are now being offered capacity on TI's satellite channel.

As with the satellites, access to 4GLs used to be very difficult, involving long bureaucratic procedures in order to import them. Because of this, only two software companies were using 4GLs prior to 1986. Following the liberalisation of software imports in the 1986 policy, access became much easier. It was made clear from early on that this was an aim of the policy. In 1984, a senior DoE official stated, 'The government will liberally allow imports of software tools which will generate more software. Import of software tools is in the interest of this country because it will generate more software and bring about productivity' (EEPC 1985). Since 1986, Indian distributors have been found for all the major 4GLs including Oracle, Ingres, Focus and Informix, so that these tools are available to Indian companies at rupee (rather than foreign exchange) prices and with some local support.

5. Conclusions

General

It is worth repeating the need to keep all of this in proportion. The software industry in India is still small, though growing fast. Within software development, a minority of contracts undertaken by Indian firms involves use of satellites or 4GLs, though the number is increasing. The case study itself has a number of features by which it differs from other studies on IDL - the two technologies considered are rather different, and the internationalisation of software has occurred later than that of previous examples; the sector is growing strongly though largely without direct foreign investment; contracting to the Third World seems to occur because of the skills shortage in the West as well as because of low labour costs; and much of the work contracted to Indian companies is actually carried out in the First World because factors like trust are also important. Nevertheless, the findings of this case study are relevant to the IDL debate.

The study reinforces the point made in the later literature that a number of factors other than labour cost are important. Trust, interaction, and hardware availability all affected decisions about body shopping, and access to local markets and government incentives had some impact on encouraging production to come to India. Above all, the issue of skills is important. The skills required for software production are such that only a small proportion of the population can take part in this production. This is one of the causes of the current skills/personnel shortage. This shortage has served to ameliorate problems, such as rising labour costs or automation of production, that might otherwise have damaged software production in the Third World.

Skill division of labour

The production process described here is subject to a skill division of labour such that the developing country more frequently undertakes the less skilled elements of production, while the developed countries do the more skilled tasks. However, this is all relative - even the least skilled tasks within software production require graduate-level employees. Secondly, the picture is not a static one, and Indian firms are increasingly being entrusted with the more highly skilled production stages, as they build up a relationship with their clients in the West.

The question of impact of technology on the division of labour needs to be treated with caution. There are many factors other than just the technologies which have an effect on IDL, such as continued high growth and skills shortages. Thus the technologies do not have simple impact on the division of labour. Instead, they tend to present opportunities which may or may not be taken up. Looking at the relationship between the two new technologies and the skill division of labour there is no obvious and dramatic impact discernible. However, 4GLs particularly do appear more likely to improve, rather than harm, the chances that more skilled tasks will be contracted to Indian firms.

Locational division of labour/'body shopping'

One of the unusual features of the internationalisation of production in the software industry is that work contracted out to Indian firms is, more often than not, undertaken outside India, at the Western client's home site. As with the skill division of labour, this is changing. As Indian firms build up their client relationships, they are able to bring more work offshore, thus avoiding 'body shopping'.

The two new technologies differ in their impacts on this process. Satellites quite clearly reduce the need for body shopping and allow more work to be brought offshore. 4GLs are yet to present any consistent impact in either increasing or decreasing the extent of offshore production.

Reverse location of production

Both technologies are automating certain processes within software development. One may therefore examine whether they are contributing to a reverse location of production from the Third World back to the First World as certain parts of the IDL literature would suggest. In this case, the concept of reverse location of production is made more complex by

the existence of body shopping. The conclusion on this was that these technologies do not appear to be clearly associated with an increase in the amount of production done in the First World.

As regards the general extent of work being contracted out to Indian companies, its rate of growth is high, and is higher than the overall rate of growth of the global trade in software. In other words, India's share of the global market is still increasing. Both technologies seem to improve the chances that work will come to India, and the converse is even more true - without these new technologies, Indian firms will lose contracts.

In the early 1980s some saw automation as 'bad' because it would lead to comparative advantage reversal and reverse location of production. Now, Ernst (1985, p336) notes that 'in offshore sites the levels of automation have been rapidly increasing', and Elson (1988, p282) states that automation is being used by countries like the NICs to 'maintain their competitive position'. The new automating technologies are now seen as something that Third World nations can and should invest in to prevent rather than cause a loss of production. To reject them would be foolhardy.

The evidence here does not necessarily refute the argument that full comparative advantage reversal will lead to a reverse location of work back to the First World. Neither satellites nor 4GLs bring about comparative advantage reversal. Satellites only make the comparative advantage more accessible, while 4GLs reduce but by no means reverse it.

What the evidence here does say is that new, automating technologies can be used in internationalised production without reversing comparative advantage and without, as yet, reversing location of production. But there are caveats. The technology is only one factor among many in decisions about location of production. For example, skills shortages in the West help this type of production to be contracted out to India. What happens in the longer term is also not very clear and there do seem to be some dangers with 4GLs. Nevertheless, this new example and its conclusions seem unsupportive of crude notions about new technology and automation leading to reverse location of production back to the First World.

There are non-technology arguments that favour a reverse location of production, but these, too, do not point to any likelihood of such a move in the medium term. Protectionism is starting to be seen in relation to software, but the action has so far been limited. Arguments

over quick response to market demand do apply to software packages, and are one of the reasons why package production for export has not taken off in the Third World. However, such arguments are much less important when undertaking the customisation and conversion work for a single customer that makes up the vast majority of India's exports. Finally, there was the question of linkages to other industries and services. The software industry requires comparatively few of these. However, such as there are - hardware, infrastructure (especially electricity), and skilled labour - do cause a problem for Third World countries. All of these problems are diminishing over time but, while they last, they are solved by body shopping which allows work to continue to be contracted out to Indian firms, while still locating production in the West.

Even though the lack of direct foreign investment means that relocation would be virtually cost free in many cases, one may conclude that currently the factors favouring internationalisation of production are stronger than those which favour a reverse location of software work back to the First World.

Generalising the case study

The unique features of this case study can make it hard to generalise from, and evidence from other industries about automation and reverse location is not so certain, possibly because these industries have reached a more advanced stage of production automation than software. Thus, while recent arguments about automation and reverse location have been tempered, they are still somewhat pessimistic about the prospects for automation and the Third World:

> "First, there is the problem of comparative advantage reversal in which manufactured goods which had previously been produced in developing countries with low labour-cost come to be produced with capital-intensive, electronics-based automation technologies in the industrially advanced economies. There is not as yet a great deal of evidence that this phenomenon has become widespread in industries *relocating* production back to the industrialised economies, outside of the electronics sector itself in which there has been some marginal shift of assembly operations back to the United States and Europe. More relevantly though - and difficult to document - it seems likely that the process of shipping out production from the industrially advanced economies to less developed ones has been slowed, and in some cases even halted." Kaplinsky (1987, p145)

It has also been suggested that the processes of relocation and international division of labour may in some ways be cyclical, or at least may flow to-and-fro depending on the state of a number of variables. Ernst (1985) interprets the processes by seeing the production

facilities in the Third World as a buffer against fluctuations in demand, which could be closed down or under-used during times of 'famine', and geared or set up again during times of 'feast'. Kaplinsky (1984, p164) also thinks that locational decisions will swing back and forth between First and Third World depending on the weight of a number of variables such as labour cost, labour intensity, boom or slump, and degree of protectionism. Because of this, Kaplinsky (1984, p167) sees possibilities for both a continued international division of labour and a reverse location of production back to the First World. More time will be needed before one can tell if the evidence from the international software industry supports these arguments.

It has been held that the factors applicable to the four Asian Tigers which allowed them to 'take off' economically and take advantage of exports and internationalisation of production, are not repeatable, and were a case of the right countries doing the right thing at the right time. 'The same conditions that allowed the rapid growth of the NICs *do not now exist* for other Third World countries' (Wield and Rhodes 1988, p309). Elson (Open University 1983b, p101) has termed these conditions a 'conjuncture' between the 'push' of a rapidly growing First World market and the 'pull' of that Third World nation's industry turning itself to export orientation. This case study would suggest that the conditions may continue to exist - but in different sectors at different times.

Such growth as is seen in the Indian industry seems to have arisen from a fortunate coincidence of a pre-existing educational and computing infrastructure, state-encouraged export orientation, and high costs and growth in the West, together with factors such as language and a wealth of Indo-American contacts. It seems unlikely that many other Third World countries will be able to follow India's example in software and one must always bear in mind the 'fallacy of composition' - whilst it is possible to conclude that software exporting may be beneficial to a single Third World nation, or even to a handful, this is not a finding that can legitimately or logically be extended to the whole of the Third World.

It would appear feasible that nations other than India are going to follow a similar path. Brazil, Singapore, Taiwan and others all have software industries which are exporting and which have certain features resembling those of the Indian industry. However, it is likely to be similar nations that took up the challenge of the IDL in the 1960s and 1970s which are also going to benefit from the IDL within the software sector in the 1980s and 1990s.

The evidence from the Indian software industry may support a conclusion that the argument has now moved on from a crude fear that new technologies will cause a reverse location of

production, and one can agree with a wider application of Ernst's statement (1985, p334) that:

> "The problem is not so much related to the possibility of outright *relocation* of industrial activities from the South....back to the OECD region. Rather, it is that the use of computerized automation technologies in electronics manufacture is likely to penetrate into a very select group of Third World industrial growth poles in Asia, Latin America and the Middle East."

Wield and Rhodes (1988, p308) note Kaplinsky's more generalised conclusion: 'He believes that a few NICs may be able to make the necessary investment in automation technologies to undermine this reversal but, overall, he is pessimistic of the chances for the vast majority of the Third World'.

References

BusinessIndia (1988) Committee to look into software exports. BusinessIndia, 8-21/2/88, p14

Confederation of Engineering Industry (CEI) (1987) Export Strategy Paper - Software. Confederation of Engineering Industry, New Delhi

Computing (1987) Quickbuild staff get rich quick. Computing, 5/11/87, p2

Cornes, Ralph (1987) The under-employed fourth generation. Computer Guardian, 20/8/87

Engineering Export Promotion Council (EEPC) (1985) Proceedings of the EEPC Workshop on Computer Software, Dec 1984. Engineering Export Promotion Council, New Delhi

Elson, Diane (1988) Dominance and dependency in the world economy. In: Survival and Change in the Third World, B. Crow & M. Thorpe (eds), 264-287. Polity Press, Cambridge

Ernst, Dieter (1985) Automation and the worldwide restructuring of the electronics industry: strategic implications for developing countries. World Development, 13(3), 333-352

Financial Times (FT) (1987) Computer services. Financial Times, 15/10/87, Special supplement

Foremski, Tom (1988) The decline and fall of US dp jobs. Computing, 9/6/88, p18

Frank, A.G. (1981) Crisis in the Third World. Heinemann, London

Government of India (1987) Software India 1987: Indo-US Conference. Department of Electronics, New Delhi

International Data Corporation (IDC) (1987) Directions '88 Vols I & II. International Data Corporation, New Delhi

Kaplinsky, Raphael (1984) Automation: the Technology and Society. Longman, Harlow, Essex

Kaplinsky, Raphael (1987) Micro-electronics and Employment Revisited: A Review. ILO, Geneva

Kumar, Arun (1987) Software policy: where are we headed? Economic & Political Weekly, 22(7), 14/2/87, 290-294

Lamb, John (1988) Exploiting Third World countries for cheap labour. Computing, 21/4/88, 24-25

Open University (1983a) Dimensions of world integration. In: U204 -Third World Studies, Block 4, Part A. Open University Press, Milton Keynes

Open University (1983b) Dominance and dependency. In: U204 - Third World Studies, Block 4, Part C. Open University Press, Milton Keynes

Poe, Robert (1987) India's soft hopes. Datamation, 1/9/87, 96.5-96.12

Rada, Juan (1980a) Microelectronics, information technology and its effects on developing countries. In: The Socio-economic Impact of Microelectronics, J. Berting, S.C. Mills, H. Wintersberger (eds), 101-146. Pergamon Press, Oxford

Rada, Juan (1980b) The Impact of Microelectronics. ILO, Geneva

Rada, Juan (1983) Information technology and the Third World. Paper presented at IFAC seminar, Vienna, March 1983

Raghavan, N. (1988) New software Mecca? BusinessWorld, 15-28/2/88, 13-14

Sauvant, Karl P. (1986) Trade and Foreign Direct Investment in Data Services. Westview Press, Boulder, Colorado

Schofield, Jack (1987) Who's afraid of a 4GL? Computer Guardian, 23/7/87, p17

Schware, Robert (1987) Software industry development in the Third World: policy guidelines, institutional options, and constraints. World Development, 15(10/11), 1249-1267

Sweet, Pat (1988) Software tools may trigger a spending spree. Computing, 11/2/88, 22-23

Thomas, Ray (1987) Users talk on up. Computer Guardian, 10/9/87

Van Themaat, J.V. & Stevens, C. (1987) The division of labour between Europe and the Third World. In: Europe and the International Division of Labour, C. Stevens & J.V. van Themaat (eds), 1-14. Hodder & Stoughton, London

Wield, D. & Rhodes, E. (1988) Divisions of labour or labour divided?. In: Survival and Change in the Third World, B. Crow & M. Thorpe (eds), 288-309. Polity Press, Cambridge

The Development Policy and Practice research group was set up in the Open University towards the end of 1984 to promote research on development issues. Its members have a wide range of disciplinary backgrounds (engineering, sociology, economics, education and geography). At present, research is focussed in three areas: food markets - particularly in sub-Saharan Africa and South Asia; the development of finance and banking; and links between small and large scale production.

DPP is a relatively small research group with limited funding. In order to increase our efficacy we are keen to enter into collaborative arrangements with other groups and development agencies where appropriate. DPP will also be acting as a centre to focus the development concerns of the Open University by arranging seminars and workshops. DPP can be contacted at the following address:

Development Policy and Practice
Technology Faculty
The Open University
Walton Hall
Milton Keynes MK7 6AA
United Kingdom

Telephone: 0908 652103

DEVELOPMENT POLICY AND PRACTICE:

WORKING PAPERS

DPP Working Papers are available either to exchange with journals or with other paper series, or on payment of £3.50 per paper. Please make cheques or international money orders payable to "The Open University (Development Policy and Practice)".

Papers marked by an * have been published in journal or book form, and are no longer distributed as working papers.

Paper No.	Author	Title
1*	M Mackintosh	Agricultural marketing and socialist accumulation: a case study of maize marketing in Mozambique
2	L Harris	Finance and money with underdeveloped banking
3*	H Bernstein	Capitalism and Petty Commodity Production
4*	B Crow	US policies in Bangladesh: the making and the breaking of famine
5*	M Mamdani	Extreme but not exceptional: towards an analysis of the agrarian question in Uganda.
6*	B Crow	Plain tales from the rice trade: indications of vertical integration in foodgrain markets in Bangladesh
7	T Painter	Migrations, social reproduction, and development in Africa: critical notes from a case study in the West African Sahel
8	N Amin	Characteristics of the international rice markets
9*	M Mackintosh & M Wuyts	Accumulation, social services and socialist transition in the Third World: reflections on decentralised planning based on Mozambican experience
10	P Woodhouse	The Green Revolution and food security in Africa: issues in research and technology development

11	N Amin	Maize production, distribution policy and the problem of food security in Zimbabwe's communal areas
12	A Akçay	From landlordism to capitalism in Turkish agriculture
13	T Evans	Economic policy and social transition in revolutionary Nicaragua
14	L Harris	Theories of finance and the Third World
15	T Hewitt	Skilled labour in R&D: a case study of the Brazilian computer industry
16	H Bernstein	Agricultural 'modernisation' in the era of structural adjustment