Individual-level evidence of IT skill specialization and formation in the advanced industrial economies[*]

Mark Huberty[†]

January 6, 2012

1 Introduction

I present new evidence of comparative patterns of specialization and technology adoption in the advanced industrial economies.

2 Institutions and skill development

Theories of comparative political economy predict that institutional variation in the advanced industrial economies will create diverging patterns of skill formation in workers, and in technology adoption and exploitation among firms. The Varieties of Capitalism literature in particular (Hall and Soskice, 2001) makes two predictions about technology and skills: first, that workers in more liberal economies will adopt more general skills than their counterparts in coordinated economies; and second, that firms in liberal economies will be more likely to engage in radical innovation and adopt leading-edge technology.

These claims rest on an implicit microeconomic model of firm and worker behavior. In more liberal economies, the lack of employment protection laws means that workers

[*] DRAFT NOT FOR DISTRIBUTION

[†] Travers Department of Political Science, University of California, Berkeley. Contact: markhuberty@berkeley.edu.

must consider the possibility of losing their jobs without recourse. Likewise, firms must consider that workers who have no protection will feel little loyalty to any given firm. In that context, both workers and firms will converge on an equilibrium that values general skills–the workers, so that they can easily find jobs at other firms, and the firms, so that workers are easily replaced. Conversely, countries with strong employment laws will tend to favor specific skills, because firms must extract high value from employees that cannot be easily dismissed, and because employees believe that investment in firm-specific skills will be rewarded by long employment tenure.

Empricially, the evidence used to support these claims has typically relied on macro-level measures of skill formation, including rates university versus vocational training, patenting rates across different kinds of technology, and aggregate R&D spending.[1] But these metrics do not directly measure worker-level behavior. Rather, they measure aggregate behavior at the societal level, and may not adapt to new markets or sectors that have different needs for employment.

3 Pursuing new data for measurement

I exploit a new dataset on technology adoption and skill formation in the information technology industry. That dataset provides individual-level evidence of cross-sectional and intertemporal patterns of skill formation among workers worldwide, and augments that data with measures of expertise and evidence of patterns of worker interaction.

StackExchange represents a collection of community-created and maintained internet sites that field questions and answers on topics related to information technology. Begun in 2008 by Joel Spolsky and colleagues, it has grown into one of the most comprehensive communities of technical information in the world. As of 2011, it reported approximately

[1] For a full set of metrics, see Hall and Gingerich (2009).

725,000 users, 1.2 million site visits per day, and 4.2 million separate answers covering 82% of the 1.9 million submitted questions. It had grown from its initial site, StackOverflow[2], to sites covering everything from programming to data security to cooking and philosophy.

The format of StackExchange interactions can be stylized as follows: a user poses a question and tags the question with metadata related to the specific technologies or technological domains the question pertains to. Other users respond to the question. Those responses, in turn, are rated by respondents, the person who posed the original question, and other community members, on the basis of accuracy, solution elegance, and completeness. Questions and answers can be retrieved by querying metadata, the user ID of the questioner or answerer, and other boolean search terms.

Users contribute to StackExchange through user accounts that contain a range of information about the user. Users may, but are not required to, provide information about their geographic location, interests, website address, and other personal characteristics. There is no evidence that this data is validated, and avatars or similarly obscured or stylized user names and identities are permitted.

All StackExchange data is collected and published under a Creative Commons license. In keeping with the license, StackExchange releases a bi-monthly dump of its entire dataset, including the full record of questions and answers, user profiles, record tagging and metadata, and time stamps. The data are anonymized with respect to user names but not other user metadata, including geographic location, web site, the questions posed or answered by each user, and the user's community-determined "reputation".[3]

[2] In programming parlance, a stack overflow refers to a program condition in which memory usage exceeds memory allocation, causing a program crash. Hence StackOverflow is, in a sense, excess memory for humans.

[3] For more information on the StackExchange data dump, see http://blog.stackoverflow.com/category/cc-wiki-dump/. For information on the terms of use for the Creative Commons license, see http://creativecommons.org/licenses/. For academic use of the data, including stud-

These data therefore represent a rich trove of information about the interaction of IT workers worldwide. The dataset contains not only the location and other information about the community members themselves, but also, through the information on questions posed and answered and their reputation, the kinds of technology they work with and are expert in, and their rating by a community of their peers. As such, the data provide an opportunity to directly observe patterns of specialization of knowledge and interaction among IT workers across both countries and time.

3.1 Other data needed

- Data to quantify relative rates of participation by nationality on Stackoverflow:
  - Workforce size by country (OECD)
  - High-technology workforce size by country (This appears difficult to get; the OECD publishes data on "ICT workforce as share of total", for both a narrow and a wide definition of the ICT workforce; we might have to back it out using those numbers plus total workforce size. See http://www.oecd.org/document/23/0,3746,en_2649_34449_33987543_1_1_1_1,00.html
- Data needed to quantify overall sector size by various measures:
  - High-technology sector size by country
  - Firm count in high-technology sectors by country
  - Startup count in high-technology sectors by country (see the venture capital associations here, I have the EVCA data)
  - ICT patents (see OECD, Patents by Technology, which has an ICT category)

ies of social interaction and exchange markets, see http://blog.stackoverflow.com/2010/05/academic-papers-using-stack-overflow-data/.

4 Research Design

To employ the StackOverflow data in the study of skill and task specificity, we first need to define exactly what is meant by the terms task and skill, and how specificity will be defined.

4.1 Definitions

4.1.1 Skill specificity

Skills will refer to knowledge of the workings of specific technologies, such as the C or SCADA languages themselves. Just as someone might be very knowledgeable about fuel injection systems, but not necessarily ever need to rebuild them, so could an IT worker be very knowledgeable about the C programming language without every using it to, say, optimize highly parallelized programs used for scientific computing. Thus we should treat skill separately from tasks.

We may infer skill specificity from a question-and-answer site like StackOverflow by analyzing the co-occurrance of technologies by question. If questions about some technology often reference one or many other technologies, we may infer that this technology is a more *general* technology than one for which questions ask about it alone. Thus, for instance, the C programming language is a general-purpose technology used for everything from commercial software development to scientific computing. In contrast, the SCADA language is used predominately for control of industrial machinery.

To quantify this definition of specificity, we must first define the relationship of technologies to each other. We can quantify this relationship as the proximity between any two technologies in the question space. Each question $q_i, i \in [1...n]$ considers some technologies T_q, which are a subset of the entire set of technologies T_Q considered in all questions

Q. The proximity between any two technologies T_i and $T_j \in T_Q$ can be defined as the conditional probability that they appear in questions together. Formally, $P(T_i \in T_q | T_j \in T_q)$ can be defined as:

$$p_{i,j} = \frac{\sum_{q \in Q} T_i \in T_q | T_j \in T_q}{\sum_{q \in Q} T_j \in T_q} \tag{1}$$

4.1.2 Task proximity

Tasks will refer to the application of technologies. Consistent with the definition in equation 1, we will begin to define task specificity by first defining task proximity. Task proximity is the conditional probability of co-occurrance of tasks in the question metadata. Thus, for instance, if questions regarding thread optimization also deal with memory optimization, but rarely deal with web page design, then we would regard thread and memory optimization as more proximate tasks than thread optimization and web page design. Using the same notation as above,

Each question $q_i, i \in [1...n]$ considers some tasks K_q, which are a subset of the entire set of tasks K_Q considered in all questions Q. The proximity between any two tasks K_i and $K_j \in K_Q$ can be defined as the conditional probability that they appear in questions together. Formally, $P(K_i \in K_q | K_j \in K_q)$ can be defined as:

$$\frac{\sum_{q \in Q} K_i \in K_q | K_j \in K_q}{\sum_{q \in Q} K_j \in K_q} \tag{2}$$

4.1.3 From proximity to specificity

Scoring tasks or skills as more or less specific requires that we translate from proximity between pairs of tasks or skills to a overall measure of specificity. Given a proximity matrix P that contains the pairwise proximities $p_{i,j}$ between all technologies (skills) $T_i, T_j \in T_Q$, we may define the specificity of any technology (skill) T_i as in equation 3, where $\delta \in (0, 1]$ is some threshold proximity value.

$$ST_i = \sum_{q \in Q} p_{i,q} > \delta \tag{3}$$

Higher-valued ST_i indicate skills that are more general, defined as applied in concert with a larger set of other skills. We can write the same relationship for task specificity SK_i.

4.1.4 Task specificity and technological clustering

Observers will note that some technologies (skills) are heirarchical. For instance, the `bash` shell scripting environment is a subset of the Unix operating system. Thus $P(unix|bash)$ will be close to 1. In aggregate, leaving this fact unaccounted for would simply reproduce heirarchies like this. To account for this problem, we take the minimum of the pairwise conditional probabilities for each skill or task pair. Thus while $P(Unix, bash)$ may be 1, $P(bash, Unix)$ will most likely be far smaller, reflecting the fact that only a portion of questions about Unix will inquire about the `bash` environment.

Alternatively, we can view this heirarchy of technologies as a natural clustering of technology types. Bridges between the technology types consist of nodes describing tasks undertaken in either technology. Thus, for instance, we may observe a cluster of "web technologies" that are separate from "operating systems" and "machine control". They may each be joined by a common node "memory management". Viewed in this fashion, users adopt skills in a technology cluster, rather than a set of atomistic technologies. A user's skill specificity is then measured as the number of clusters that he or she specializes in.

4.1.5 Individual-level skill and task specialization

Given these definitions of task and skill specificity, we can now define what we mean by those terms for any given individual. for a user $u \in U$, their skill specificity is the mean specificity in all questions $q \in Q$ that they have answered. Task specificity is defined the same way. Each user thus receives both a skill and a task specificity score. Conceptually, then, each user falls into one of four possible categories: tackling specific tasks with general skills, general tasks with general skills, specific tasks with specific skills, or specific tasks with general skills. Notice, here, however, that we are not bound by discrete categorization. Rather, we can measure both task and skill specialization as continuous quantities.

Using the alternative definition of task proximity, we can define specialization as the number of different technology clusters in which a given user displays expertise. This eliminates the (potentially artificial) distinction between tasks and skills.

4.2 Technical design

4.2.1 Option 1: continuous measures of tasks and skills

1. For the full set of users U, select those with geo-coded information in their profiles, U_g (DONE)
2. For U_g, disambiguate the geo-coded data (DONE, via the Yahoo Placemaker service)
3. For all questions Q, get two subsets of questions:
    (a) All questions for which the submitter and AT LEAST one answerer are in U_g
    (b) All questions for which AT LEAST one answerer are in U_g

4. Get all tags T in the question set Q. Subset for only those tags T^* that occur more than N times in the dataset. Construct a proximity matrix M_{T^*} with the conditional probability of tag co-occurrance as defined in equation 1

5. For all tags T^*, categorize as either skills or tasks

6. For all users U_g, get all tags in all questions answered by U_g and the community-generated score of the answer. Compute the weighted task and skill specificity measures as described in equation 3 and section 4.1.5

4.2.2 Option 2: clusters of tags

1. Get the users and tags as described in section 4.2.1.

2. For tags T, construct the graph G_t of tags wherein each tag is a node and the edges are defined as the conditional probability of co-occurrance. Only use edges for which $p > threshold$.

3. For G_t, find the graph k-cores (see `http://networkx.lanl.gov/reference/algorithms.core.html` and associated references).

4. For each k-core, get the tags associated with it

5. For each user U_g, determine which tags $T' \in T^*$ apply to their answers

6. Count the number of tags they answer, in the number of k-cores, for each user in U_g

5 Hypotheses

5.1 Approach 1: continuous measures

H1 Users from CME countries will have higher mean specificity, but in fewer skills or tasks (i.e., will display strong but relatively narrow expertise)

H2 Users from LME countries will have lower mean specificity scores, but will display expertise in more skills or tasks (i.e., will display weaker but broader expertise)

5.2 Approach 2: clustered measures

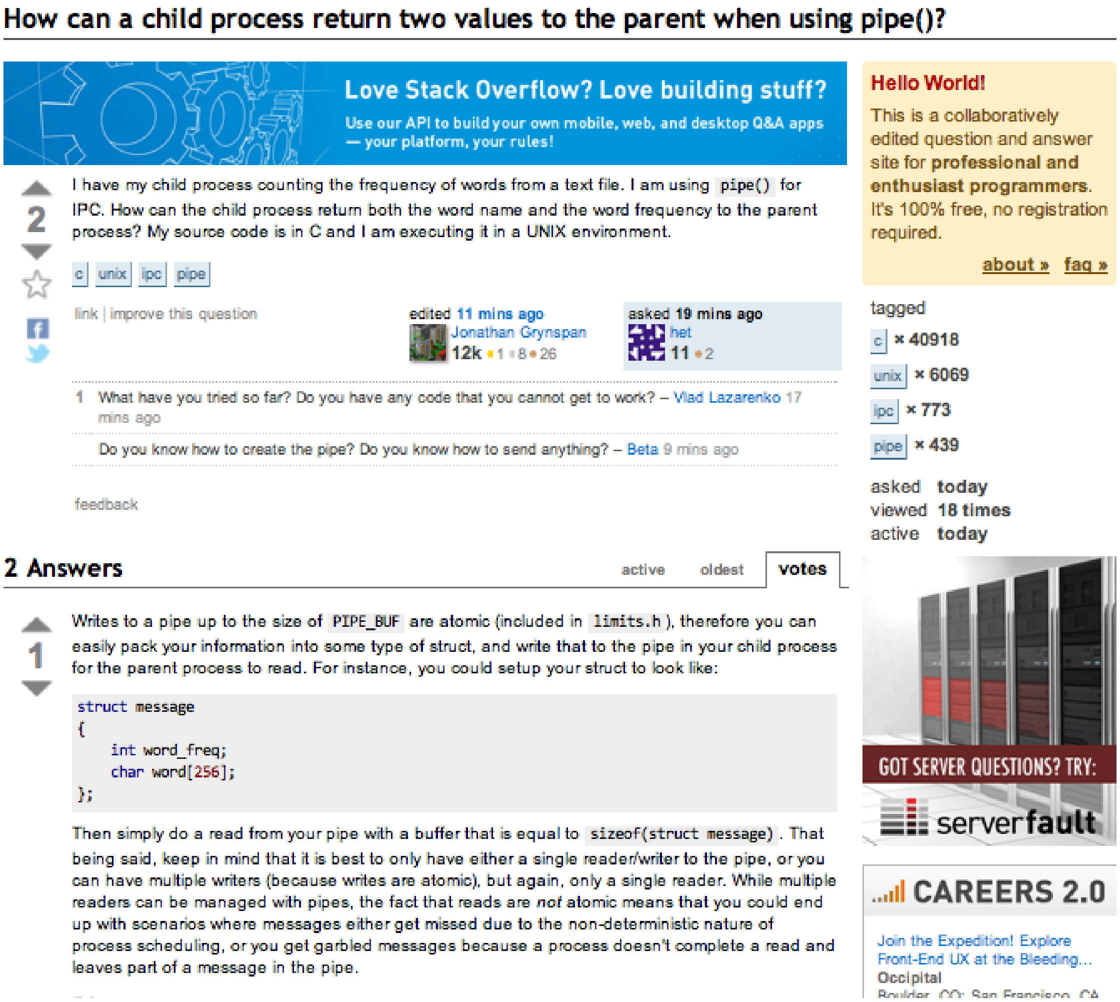
H1 Users from LME countries will display expertise in more technology "clusters" than users from CME countries. This is equivalent to saying that LME users will have a more general skill set, as a function of more liquid labor markets.

H2 Users from CME countries will display expertise in more tags in any given cluster than users from LME countries, even if they are present in fewer technology clusters

6 Results

7 Discussion

8 Conclusions



How can a child process return two values to the parent when using pipe()?

Love Stack Overflow? Love building stuff?
Use our API to build your own mobile, web, and desktop Q&A apps — your platform, your rules!

2

I have my child process counting the frequency of words from a text file. I am using `pipe()` for IPC. How can the child process return both the word name and the word frequency to the parent process? My source code is in C and I am executing it in a UNIX environment.

c unix ipc pipe

link | improve this question

edited 11 mins ago
Jonathan Grynspan
12k ●1 ●8 ●26

asked 19 mins ago
het
11 ●2

1 What have you tried so far? Do you have any code that you cannot get to work? – Vlad Lazarenko 17 mins ago

Do you know how to create the pipe? Do you know how to send anything? – Beta 9 mins ago

feedback

Hello World!
This is a collaboratively edited question and answer site for **professional and enthusiast programmers**. It's 100% free, no registration required.

about » faq »

tagged
c × 40918
unix × 6069
ipc × 773
pipe × 439

asked today
viewed 18 times
active today

GOT SERVER QUESTIONS? TRY:
serverfault

CAREERS 2.0
Join the Expedition! Explore Front-End UX at the Bleeding...
Occipital
Boulder, CO; San Francisco, CA

2 Answers

active oldest votes

1

Writes to a pipe up to the size of `PIPE_BUF` are atomic (included in `limits.h`), therefore you can easily pack your information into some type of struct, and write that to the pipe in your child process for the parent process to read. For instance, you could setup your struct to look like:

```
struct message
{
    int word_freq;
    char word[256];
};
```

Then simply do a read from your pipe with a buffer that is equal to `sizeof(struct message)`. That being said, keep in mind that it is best to only have either a single reader/writer to the pipe, or you can have multiple writers (because writes are atomic), but again, only a single reader. While multiple readers can be managed with pipes, the fact that reads are *not* atomic means that you could end up with scenarios where messages either get missed due to the non-deterministic nature of process scheduling, or you get garbled messages because a process doesn't complete a read and leaves part of a message in the pipe.

Figure 1: Sample StackOverflow question and answer

References

Hall, P. and Gingerich, D. (2009). Varieties of capitalism and institutional complementarities in the political economy: An empirical analysis. *British Journal of Political Science*, 39(03):449–482.

Hall, P. and Soskice, D. (2001). *The Varieties of Capitalism*. Oxford University Press, Oxford.