Requirements Engineering in the Long-Term: Fifty Years of Telephony Feature Evolution

Annie I. Antón
College of Engineering
North Carolina State University
1010 Main Campus Drive (EGRC 408)
Raleigh, NC 27695-7534

anton@csc.ncsu.edu

Colin Potts
College of Computing
Georgia Institute of Technology
801 Atlantic Drive
Atlanta GA 30332-0280

potts@cc.gatech.edu

Abstract: Systems are useful to the extent that the potential benefit they provide customers outweigh the responsibility costs that customers incur to realize those benefits. We develop a classification of benefits (potentials) and costs (responsibilities) in the domain of interpersonal communication features. Using this classification, we chart the introduction and availability of telephony features to private telephone subscribers over a fifty-year period in a metropolitan area. The growth of features occurred in bursts, emphasized different potentials over time, and they imposed burdens on customers that increased with the power of the features themselves. A punctuated equilibrium model of evolution explains discontinuities in the introduction of features, the displacement of some older features by newer ones, and the dynamics of the cost/benefit tradeoffs that subscribers experience. We explain the non-uniform introduction of types of feature over time in terms of the cultural and technological context of telephony use. In particular, we identify three expansion epochs in which the major concerns were communication, privacy, and accessibility and awareness. We conclude by discussing the implications of our data for requirements engineering in the context of long-lived, feature-rich multi-agent systems.

I. Introduction

Requirements engineering principles and practices are designed to assure the appropriateness of system features. In systems with many users or customers, requirements engineers cannot assess the appropriateness of proposed features by simply asking customers what they want [Bey98, Pot95a]. Since such systems tend to be E-type systems [Leh80, LB85] that continue evolving to adapt to changes in their environment, new features always appear in the context of existing features, modes of use, enabling technologies, and past architectural decisions.

Features are complex systems of required and assumed behaviors. The cohesiveness of these behaviors determines how the feature fits into and shapes its environment and how it interacts with other features or ways of working. An understanding of how features evolve would therefore benefit requirements planning and validation. Some authors also suggest that services appear directly in the implementation architecture as application-layer subsystems [Ben97], a strategy that could provide an avenue for requirements-based system composition and aggregation.

Whether a feature is appropriate is a complex judgment. Appropriateness includes the traditional criteria of requirements completeness, consistency, absence of gold-plating, unambiguity and feasibility, and there is an active research literature in the assignment and management of priorities to requirements [BI96, Kar96, KR96]. These criteria and trade-off techniques, however, apply at the level of individual requirements or their interactions. They do not address the overall cohesiveness of features or their convenience and desirability for customer activities.

We need to be able to summarize succinctly what a feature is without having to specify it in detail, but with greater clarity and precision than is afforded by such phrases as “voicemail”, “call waiting” or “footnotes.” And we need better ways to categorize and roughly quantify the predicted benefits of proposed features so that they can be weighed against the similarly categorized and roughly quantified costs.

The first question, that of determining feature boundaries, or modularizing requirements into logically coherent bundles, can be approached in a number of ways. One approach would be to define, as a feature, anything that the marketing literature defines as such. Thus “call-forwarding” or “caller ID” would qualify as telephone features if subscriber-oriented feature guides or advertising literature mentioned them. But then, “Y2k compliant” or “cheaper than the competition” would qualify too. What Call Forwarding and Caller ID have in common in contrast to Y2k compliance or price is that they summarize operational processes and benefits. The defining and bounding of features therefore has to involve the description of purposeful behaviors.

Another way to address features in a proposed system is to ignore the nature of the features and instead count how much feature-exhibiting potential a system possesses. In project estimation, this idea of potential is enshrined in the measuring of “function points,” weighted combinations of input- and output-attributes of a planned system [Mac94]. Function points are a more requirements-oriented basis for estimates than lines of code, but they rely on the assumption that all function points are equal. Not all features are equally valuable.

Our approach borrows from both strategies. We characterize features as being modules of benefit: They deliver potential to accomplish goals, but they also impose behavioral burdens. We analyze features as yielding a combination of different types of goal-achievement potential and imposing different types of burdens.

As for the second question, the categorization of benefits and the trade-offs among benefits and costs, we adopt an equilibrium model in which features are useful to the extent that the potential benefits they provide customers outweigh the responsibility costs that customers incur to realize those benefits. Customers, as in the usage of the Soft Systems Method [CS90], are intended beneficiaries of a feature, whether they are paying customers, end users, or indirect beneficiaries. Benefits of a feature are any *potential* capabilities that a feature supplies the customer, including objectives that the feature directly helps the customer achieve. Consider VoiceMail, which supplies subscribers with potential beyond the underlying telephone service that it supplements (e.g., being able to know the identity of a person who called earlier).

Just as the benefits of a feature are the potentials that the feature reveals so its costs consist of the responsibilities, or *burden*, that the customer incurs in order to realize the feature's potentials. These costs may include equipment purchases and physical, mental or organizational work. For example, VoiceMail imposes responsibilities beyond those incurred simply due to being a telephone subscriber; subscribers have to periodically set up a greeting message and remember to check whether there are messages waiting by being near the phone if it is equipped with a message-waiting light. Many of the responsibilities incurred by subscribers increase the cognitive complexity of using the telephone, and many vocal critics of feature "bloat" or "encroachment" in software in general argue that the added complexity of further features often outweighs their power [Nor98].

In the next section, we develop a classification of potentials and responsibilities in the domain of interpersonal communication features. This is a more restricted domain than the whole of software engineering, but it is much more encompassing than a single application. Our intention is to give the requirements engineer or product designer a vocabulary for reasoning about the benefits and costs, potentials and responsibilities, and power and complexity of proposed features. For applications in other domains, a revised vocabulary will be necessary, but the approach that we discuss below is general to feature-driven systems.

While requirements engineering is concerned with the proposal and analysis of planned features, the issues raised by the two questions above are more clearly illustrated and analyzed by considering the emergence of features in existing systems. Since required features are simply hoped-for properties and are described in the future tense or optative mood [Jac95], once successfully implemented in a delivered system they may be described using the present or past tense and the indicative mood. Thus any discussion about the appropriateness, coherence, overlap or growth of required features in a planned system is formally equivalent to the corresponding discussion about the actual features of a previously implemented system. It is much easier to analyze existing systems than imaginary ones, and in Section III, we describe how telephony features for personal subscribers have evolved over a fifty-year period in a metropolitan area. We then discuss the evolutionary patterns that this E-type system has exhibited and propose how such knowledge can improve the validation and planning of future features.

II. A Framework for Analyzing Features

A feature is a bundle of behaviors that serves some useful and coherent purpose for a customer.

A. Features and Other Modeling Constructs

Features are closely related to other concepts, such as goals, architecture components, use cases, scenarios, obstacles, agents, tasks and activities.

1) Goals and the purposefulness of features

A feature involves the choreographed actions of several agents, at least one of whom is the beneficiary because the feature promotes the achievement of one of the beneficiary's goals. We judge as appropriate those features that support goals. For example, Call Forwarding supports the goal of avoiding missed calls (or, more abstractly, maintaining contact even when away from one's usual location). Those features that do not support goals or that support minor goals (at a cost that is too great to justify the feature) are deemed pointless or merely decorative. While the agents and actions of a feature are relatively unambiguous and objectively determined, its purposes are more fluid and subjective. Customers, requirements engineers and inventors predict poorly the value of proposed features, since customer goals are often unclear and volatile. Indeed, a new feature may create further demands because of accidental affordances; as in the use of answering machines to screen calls.

Previous research in requirements engineering has developed a rich theory of goals and their refinement into system constraints and operations [Ant96, Ant98, vLDM95], the obstacles that may block goals in the deployed system's environment and the secondary defensive and mitigation goals that arise to make the system more robust in the presence of such obstacles [Pot95b, Pot99]. Features are clusters of goal-achieving and obstacle-defending or obstacle-mitigating behaviors and properties. Features logically cohere in the following ways: (1) they are discrete additive functional molecules of value that cannot be decomposed arbitrarily into actions or constraints that provide partial value; (2) an agent, the customer, obtains a net benefit in terms of potential goal achievement; and (3) they may necessitate the orchestrating of other agents' behaviors to ensure that the goals are achieved.

2) Features, use cases and scenarios

Features are closely related to use cases [Jac92, Fow97]. Use cases are modes of system use that external agents initiate with a purposeful input. Although authors in the object-oriented methodology invented the concept of use cases [Jac92], use cases are less object-oriented units than they are units of purposeful interaction. Basic use cases usually use subordinate or refined use cases, and so it is with features. For example, in telephony a message-waiting indicator feature is meaningless except as a refinement of voicemail. There can be no messages waiting unless there is a voicemail feature to store them. Scenarios, the more detailed explication of the ordered steps and interactions followed in the playing out of a use case, may be used to explain features and the potential conflicts between features. Consider Figure 1, it shows the interactions between a number of internal system components and external agents during the playing out of the Caller ID Deluxe feature offered by most local telephone companies.

Features are not just use cases. Unlike use cases, a feature's scenarios may exhibit temporally disjointed episodes. For example, in the Caller ID Deluxe feature, the subscription episode, initiated by the subscriber, occurs just once. Then, after an indeterminate interval, the caller (a different person from the subscriber) initiates an identification episode, a type of episode that is repeated for every subsequent incoming call (Note that the identification need not be successful for the episode to occur.). Other features may involve more types of episode. For example, voicemail involves subscription, calling and recording a message, retrieving messages, and even, arguably, replying to them. A use-case analysis of Caller ID Deluxe would differ markedly in making the subscription and calling episodes completely different use cases. Indeed, recent accounts of use-case analysis re-cast use cases themselves as a type of "actor" outside the system with which the use case interacts [Sch98]. Here, for example, the call use case would interact with the subscription use case in exactly the same sense that it interacts with the recipient. This awkwardness in reifying subordinate use cases arises from regarding use cases as tools for suggesting object classes and designing physical components from them. The longer-term purposefulness and coherence of the episodes, so important to the customer, becomes fragmented from the architectural perspective. Requirements engineering benefits from an initial emphasis on use cases, but they benefit in turn from a semantics that connects them to purposeful activities [Pot95b, Pot99].

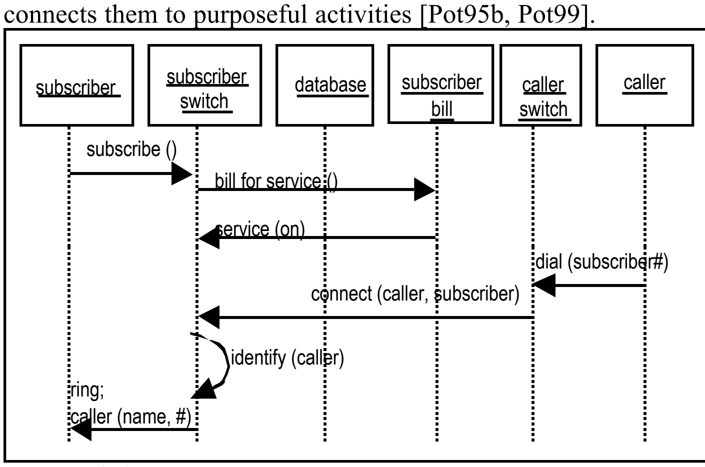


FIGURE 1: A UML INTERACTION DIAGRAM FOR THE MAINLINE SCENARIO OF CALLER ID DELUXE IN A TELEPHONY SYSTEM. (CALLER ID DELUXE DELIVERS THE CALLER'S LISTING, IF POSSIBLE, TO A CALLER-ID-EQUIPPED TELEPHONE, INCLUDING THE CALLER'S TELEPHONE NUMBER AND NAME.)

One agent always initiates each phase or episode of a feature scenario. The initiation of an episode may be the accidental by-product of the initiator pursuing other goals. For example, in the caller-id example, the caller does not deliberately initiate the calling and identification episode, because the caller's goal is merely to communicate with the recipient. The identification goal is a goal of the recipient, not the caller.

3) Involvement of subsystems in feature provision

Features always involve some automated actions in the system, and may also elucidate interactions among subsystems. In requirements engineering from the Context Diagrams of Structured Analysis [DeM79] to the formal specification of operations, writers and practitioners traditionally regard the required system as a single black-box. More recently, attention has turned to the evolution of systems whose major architectural partitions are fixed [Goe97, Pot99]. Whether a feature requires the recognition of internal system components depends on whose perspective is being adopted. The naïve telephone subscriber, for example, may not distinguish between the switch or network database subsystems and the billing system, but may acknowledge a difference between the local switch and the caller's in the case of a long-distance call. In earlier times, subscribers had to adopt different behaviors when calling a local call, a "really" local call, a long-distance call to some cities, and a long-distance call to others. Even the subscriber's external view of the system in these cases inevitably involves some decomposition into distributed components.

Features often ermerge through the incorporation of an external entity or the automation of previously external behaviors, and in these cases it is more useful to preserve the knowledge of the physical division of responsibilities, especially when things can go wrong and users must adopt workaround procedures. Specific examples of multi-component responsibilities are those that the system *mediates*. Mediation includes brokering and matching activities that are initiated by request. In telephony, they were initially provided by human operators, but sometimes now involve fully automated services that nevertheless seem separate from the core services that they mediate. For example, many features require subscribers to call a special number to activate or access a service. The number acts as an automated operator and often accesses a service with which the caller interacts through speech and dialing actions.

4) Features, obstacles and breakdowns

A feature provides a customer with the potential to achieve goals, but this potential may remain fallow for many reasons. Equipment failures or system unavailability may make the feature unavailable or degraded. Unwillingness or inability to perform an associated action, such as looking at a display panel or discriminating among several dial tones that encode information about the incoming call also undermines the value of the feature. These are *obstacles* [Pot95b, Pot99].

Features are not the only ways for a customer to achieve his or her goals. *Workarounds* are bundles of manual behaviors and the inventive uses of other automated features to achieve the same purpose, albeit often imperfectly and less conveniently. In the absence of Caller ID, a caller and recipient might agree on a code that identifies the caller, such as ringing twice in quick succession. Workarounds may be useful in the presence of feature unavailability, excessive cost, or temporary breakdown, but they typically result in degraded outcomes and require additional responsibility, including, sometimes additional actions by agents other than the customer.

What starts as a workaround may later become partly automated and "colonize" the niche of an existing feature in the customer's space of goals. Conversely, workarounds may disappear. For example, many libraries that once had card catalogs now have electronic catalogs. While the catalog system is working the electronic form is more powerful.

5) Features, Activity Theory and task analysis

In Activity Theory [Nar96] tools are an integral part of the activity, shaping the way we conceive of and plan activities and learn skilled behaviors. What constitutes a "tool" in Activity Theory is more abstract than the everyday use of the term implies, since it includes any part of the world or culture, such as language, whose affordances lend it to being recruited to some purpose. Because features affect the behavior of agents and constrain their choices, features are tools in the Activity-Theoretic sense. As activities change in the world (e.g. increased use by business of telemarketing) so the appropriateness of a feature (e.g. Caller ID) changes and the adoption of other tools and behaviors will follow as a result of these changes. A feature such as Caller ID may adapt the behaviors of a caller, as over time callers learn tactics for publicizing or hiding their identity.

B. The Benefit Profile of a Feature

Features endow customers with the potential to achieve goals that they could not achieve or could achieve less adequately in its absence. For example, Caller ID lets subscribers identify callers without answering, a knowledge acquisition goal and a subgoal in such extended goals as screening calls (see Figure 2) or preserving one's privacy from interruptions.

Potential defines the upper bound of benefit. How much benefit customers realize depends on many factors. The major impediments to benefiting from feature potential are the burdens that the feature imposes on the agents that participate in it. A feature may only be feasible if its agents possess certain capabilities. For example, Caller ID is meaningless in the context of a subscriber telephone that is not Caller ID-enabled. At the very least, it should have a display capable of showing the caller's number. Thus features may impose responsibilities on agents that they did not have before the feature was available. Burdens may involve extra setup activities, memorization or attention, or constraints on the customer's mobility or location. Features vary in their profile of burdens, but unless the customer bears them the customer gathers little benefit from the feature.

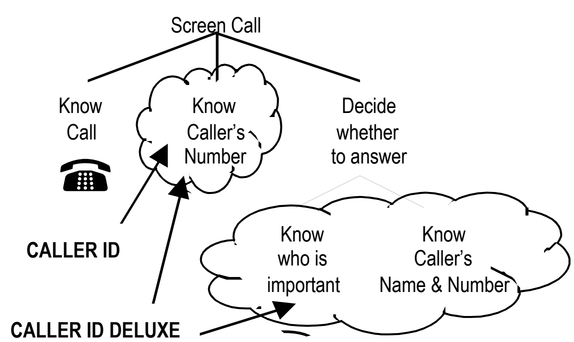


FIGURE 2: KNOWLEDGE GOALS FOR SCREEN CALLS

1) The Positive Side: Potential

Potential refers to the goals that customers can achieve with a feature that they could not achieve or could achieve less adequately in its absence. Features provide types of goal-achievement potential that we categorize as follows:

1. <u>Communication</u>: Being able to communicate information with other people beyond the potential afforded by the system's infrastructure. For example, in telephony, Three Way Calling allows customers to communicate simultaneously with two parties and the picture phone enhances the basic level of communication via additional visual richness.
2. <u>Privacy</u>: Being freer from interruption, being less likely to disclose personal information or being more autonomous in communication decisions. For example, Anonymous Call Rejection offers both freedom from interruption and privacy since subscribers will not be disturbed when a caller with Line Blocking calls.
3. <u>Organization</u>: Being able to organize and maximize cognitive and external information resources. For example, Speed Calling maximizes cognitive resources by allowing subscribers to reach frequently called numbers by dialing only one digit.
4. <u>Awareness</u>: Knowing more about the surrounding situation, including knowledge about communication events and any background contextual information used in communication decisions. Beep Tone for Recorded Conversations, which was introduced in 1948, lets a customer know through a beep tone that the conversation is being recorded. Call Waiting makes the subscriber aware that someone is calling on the "other line" and RingMaster allows the subscriber to know who is the intended recipient of a call.
5. <u>Accessibility</u>: Being accessible to others beyond the accessibility provided by the underlying service. For example, Call Forwarding Busy Line offers the potential to be accessible even when the line is busy and VoiceMail offers the potential to be available when customers are unable to answer the phone.

2) The Negative Side: Burden

The growth of features and potentials imposed burdens on customers that increased along with the power of the features themselves; we characterize these burdens as follows:

1. <u>Equipment</u>: Being responsible for purchasing or providing specialized equipment to enable a feature. With the advent of Caller ID, special Caller ID boxes and telephones with built in Caller ID support became available.
2. <u>Collocation</u>: Having to be in a specific place at a given time in the communication activity. Call Forwarding requires that the recipient or a representative is at the forwarded location .
3. <u>Action</u>: The need perform special actions during the normal span of the feature. For example, Voicemail requires the subscriber to dial into the voicemail system and perform selection operations.
4. <u>Setup</u>: Special actions needed to initiate availability of the service but not normally required operationally. These include subscription actions, programming of short cuts, etc.
5. <u>Cognitive</u>: Additional cognitive load, including short-term memory, discrimination of coded signals and decisions to act. In the case of RingMaster, subscribers must distinguish among three distinctive ring patterns that signify the recipient. Call Waiting requires subscribers to decide whether they are available.

3) Degree of automation

Customers experience different degrees of automation among features. Mediated features require some form of external intervention; for example, the assistance of a telephone operator in order to obtain the desired potential of some service (e.g., conference calls). A further distinction can be made between the types of mediation; mediated features require either human or automated mediation. *Human mediation* occurs when the operator is called to invoke some feature (as when placing a collect call). Alternatively, *automated mediation* occurs when subscribers dial a Special Access Number (SAN) to experience the benefit of a feature (e.g., voice mail and remote call forwarding). Those features that do not require the assistance of an operator or the dialing of a SAN are *non-mediated*.

C. Feature Evolution

Feature evolution is the phenomenon by which a system's features change over time. Experience suggests that the dominant form of feature evolution is the addition of new features to a *baseline*, but existing features may become refined or specialized, and obsolete features may be *displaced*. Our framework for classifying feature potentials and burdens lets us examine the dynamics of feature evolution. Are there trends in which potentials and burdens are introduced at the same time? Is feature evolution continuous characterized by relatively stable enduring baselines between sudden feature expansions? When, if ever, does displacement occur? Is feature evolution *radial*, with growth occurring uniformly across all kinds of potential and burden? That would be consistent with intuitions about feature "bloat", the view that more features are better. Or is it *tropistic*, with different potentials and burdens receiving greater emphasis at different points in the system's evolution? This would suggest directed adaptation of features based on broader concerns such as which goals are valued most at different times and are most feasible to support with features, and which burdens are acceptable to the customer community.

III. Feature Evolution in Telephony

A. How Features Evolved in Telephony

At the turn of the century, the potential for services such as call waiting and call redirection existed, but was limited to operators who had knowledge of a subscriber's movements. Obviously, the capacity and privacy (as we know it today) of this system was extremely limited. However, Almon B. Strowger's 1889 invention of the automatic telephone exchange [Emm89] resulted in a rapid fall to a minimal level of intelligence in the network; this minimal level of network "intelligence" continued until the last 20-25 years. In the mid 1960s, implementations of services were based on Stored Program Control (SPC) in each exchange within the network, facilitating the early implementation of call waiting. Interestingly, there was no real growth or expansion in the services offered during the 1970s even though SPC was used in operations systems to support and update the exchanges, providing advances in network planning and engineering. In the late 1970's the Federal Communications Commission (FCC) deregulated customer premises telephone equipment, resulting in "cosmetic" changes marked by the new availability of colorful phones and a flurry of Mickey Mouse and Snoopy phones. Perhaps this was a superficial indicator to consumers, foreshadowing the looming increase in flexibility and availability of more serious and powerful telephone services. In the 1980's, the availability of centralized databases to support phone services made it possible to manage large amounts of data in one location, leading to the introduction of the Advanced Intelligent Networking (AIN). AIN made it possible to remove the service control function from many switches to a few centralized network-resident databases. These advances in technology enabled the more rapid and consistent introduction of new services throughout the network.

B. Feature Expansion

A *feature expansion* occurs when a number of new features are introduced together, at times even displacing some older features. For example, Caller ID Deluxe, which provides both the name and number of the person calling the subscriber, displaced Caller ID, which only provided the caller's phone number. We examined the growth of telephony features over a period of 51 years (1948-1999) in a major metropolitan area. Feature growth occurred in bursts, as shown in Figure 3[1], and we characterize these bursts as periods of *feature expansion*.

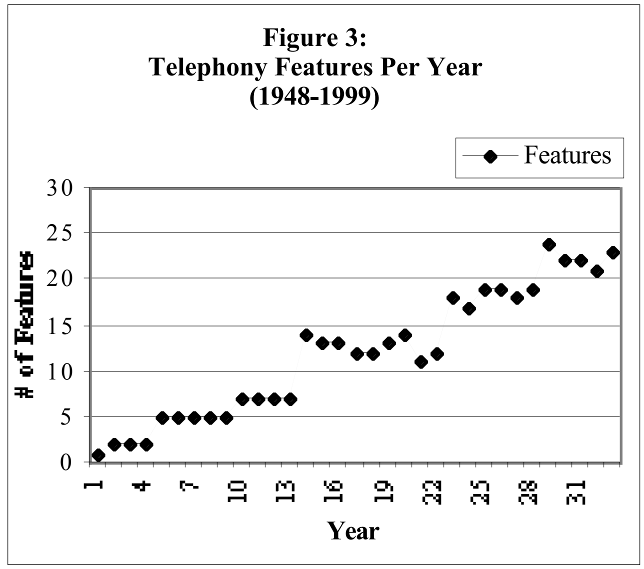


Figure 3 portrays a punctuated (non-continuous) feature growth, marked by four major feature expansions: Year 5 (1971), Year 14 (1980), Year 23 (1989) and Years 28-29 (1994-95). These four periods of expansion were each influenced by emerging technological innovations. While the expansion of 1971 is not due to the new availability of a particular enabling technology, it is significant because it marks the first clear introduction of a cluster of features. Touch-tone service was introduced in 1980 and it spurred the second major expansion during which Speed Calling and Three Way Calling were introduced.

Another distinction between features remains to be made. We distinguish between baseline, new, and displaced features as follows. A set of new features introduced before and during an expansion is referred to as a *feature cohort* and the set of features it enhances (i.e. those existing at the end of the previous expansion) is a *feature baseline*. Any features that are removed before the *next* expansion are a *displacement cohort.* The features that comprise each expansion's feature baseline, feature cohort, and displacement cohort are listed in Table 1. The introduction of features and the displacement of older features by new features are visible.

---

[1] For purposes of this paper, all charts graph 33 discrete years, but these 33 years actually span 51 years. Year 1 represents 1948 (the year in which the Beep Tone for Recorded Calls was first introduced), Year 2 represents 1961 (the year in which the Conference Call was introduced) and Year 3 represents 1967 (the first of the consecutive years for which we have more detailed data).

Expansion	Baseline	Feature Cohort	Displacement Cohort
1971	BeepTone TwoParty	*Collect* *ConfCalls* *P2P*	none
1980	BeepTone *Collect* *ConfCalls* *P2P* TwoParty	*BillTo3rd* CallFwd CallWait *PicPhone* SpeedCall *TDD* *Time&Charges* ThreeWay *TTY*	BeepTone *PicPhone* *TDD* TwoParty
1989	*BillTo3rd* CallFwd CallWait *Collect* *ConfCalls* *P2P* SpeedCall *Time&Charges* ThreeWay *TTY*	CallBlock CallReturn CallSelector CallTracing PrefCallFwd RepeatDial *RstrctCallCard* *SeqCalling* *TDD* *911*	*RstrctCallCard* *SequenceCall* *TDD* *TTY* *911*
1994-95	*BillTo3rd* CallBlock CallFwd CallReturn CallSelect CallTracing CallWait *Collect* *ConfCalls* PrefCallFwd *P2P* RepeatDialing SpeedCall ThreeWay *Time&Charges*	Block900 CallerIdListing CallerIdNmr CallFwdAll CallFwdBusy CallFwdDontAns CallFwdMain *RemoteCallFwd* Ringmaster *TTY*	*BillTo3rd* Block900 *CallerIdNmr* *CallFwdAll* *CallFwdMain* *ConfCalls* *P2P* *Time&Charges*
To present	CallBlock CallerIdListing CallFwd CallFwdBusy CallFwdDontAns CallReturn CallSelect CallTracing CallWait *Collect* PrefCallFwd *RemoteCallFwd* RepeatDialing Ringmaster SpeedCall ThreeWay *TTY*	*AnonCallReject* BeepTone CallWaitDX CustCodeRestrict FlexCallFwd LineBlock MsgWaitIndicatr *Voicemail*	none

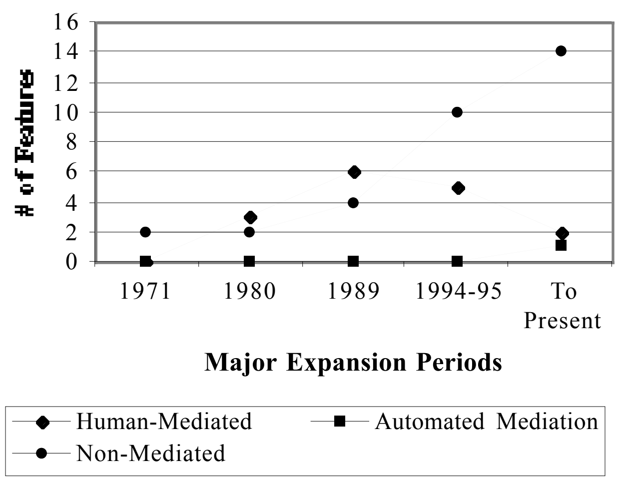
TABLE 1: FEATURE EXPANSIONS:
ITALICIZED FEATURES INDICATE MEDIATED FEATURES.

There is some evidence that potential declines immediately after an expansion. A diffusion explanation of this is that technological opportunities make many "neat" features possible, which are subsequently consolidated when it turns out that they are not all necessary or are contested on regulatory grounds. An evolutionary explanation is that some previous features are now seen to be redundant and the new ones displace them, as in the displacement of Caller ID by Caller ID Deluxe. Each new feature is supported by some degree of automation as discussed below.

C. Degree of Automation

During the early years of Plain Old Telephone Service (POTS), most special features such as long distance service required an operator. With the introduction of advanced technologies the need for operators has declined. For example, Call Waiting is "automated" since the only responsibility or burden that it imposes is the need for the customer to subscribe to the service. In contrast, placing a collect call still requires an operator, who then places the call for the customer. Table 1 distinguishes such mediated features from non-mediated features by displaying the mediated features in italics. In Figure 4, features are charted according to their degrees of automation (non-mediated as well as human and automated mediation).

Figure 4: Degree of Automation in Feature Baseline



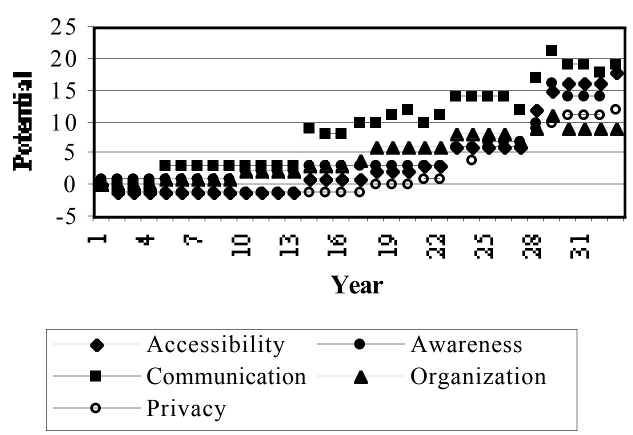
Although one would expect the degree of automation to increase over time it is interesting to note that this trend was actually slowed or even reversed by the introduction of newer, powerful features that require occasional mediation. As shown in Figure 4, the number of non-mediated features has risen dramatically since 1980. While the number of automated mediation features only began to rise during the last expansion period, the number of human mediated features is clearly declining. This may be due, in part, to the fact that many new systems initially require a staff of specially trained users or professionalize some aspects of system use. Later, however, automation tends to decentralize this expertise (disintermediation) and return responsibility to the user as evidenced by the continued growth of non-mediated features. Each feature expansion has allowed customers to become more autonomous since they are now more able to control what they do when they employ these services without having to rely on other agents to achieve their goals.

Telephony has certainly come a long way from the late 1800's when there was uncertainty as to whether customers would be able to succeed at such cognitively complex tasks as dialing a simple phone number!

D. Telephony Feature Potential

Figure 5 charts the feature potential for each of these five categories. We observed that total potential rose in accordance with the introduction of new features which enabled customers to more readily achieve their goals; thus, it stands to reason that each individual potential category experienced a similar growth. Figure 5 highlights several individual trends and three years are particularly noteworthy: Year 13 (1980), Year 22 (1989) and Year 28 (1995). In 1980, four features were introduced (Three way Calling, Call Forwarding, Call Waiting and Speed Calling) which significantly increased communication potential by providing more effective ways of getting in touch with one person. The growth in communication potential that year is attributed to the introduction of these features. A sharp increase in the potential for privacy is also illustrated in Figure 5. This is partly due to the inception of a growing concern for privacy in 1989 with the introduction of features such as Call Block, Call Selector and Call Tracing. These new features afforded customers the potential to better manage their presence and ensure that unwanted interruptions would be limited or minimized. In 1995, all five categories of potential experienced an expansion. This "across the board" increase was due to the introduction of more broad spanning features which touched all five kinds of potential.

Figure 5: Potential Categories by Year

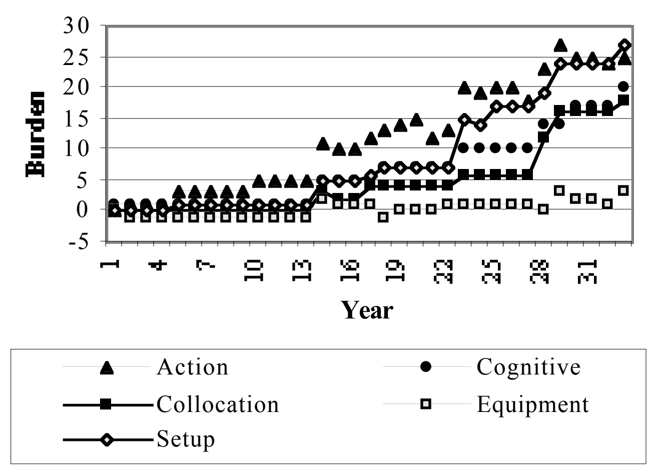


Feature expansions have been driven by a desire for more effective and convenient modes of communication, we expect this trend towards increased potential to continue, but with each increase in potential comes an increase in burden or responsibility on the part of the customer or subscriber as discussed below.

E. Telephony Feature Burden

Figure 6 charts the five kinds of feature burden per year. Action burden is fairly linear and monotonically increasing and the equipment burden is clearly not a factor in comparison to the other burdens. The cognitive burden is "bursty," but not particularly as interesting as the collocation burden and the setup burden. Collocation is a major expense because you can only achieve the associated potential if you're willing to be somewhere. There were two specific peaks in collocation burden, one in 1980 and the other in 1990. Recall that Call Forwarding was introduced in 1980 and this was the first time that customers were able to remain in communication and accessible while away from their home phone number. Of course, with this potential comes a collocation burden. For a subscriber to fully appreciate the potential of Call Forwarding, the subscriber must be physically present at the number to which their calls have been forwarded. Another feature burden, setup, also experienced peaks. Again, subscribers can only enjoy the extra power that comes with certain features at a personal expense; setup burden entails a willingness on the part of the subscriber to program certain features. In 1980 and 1989 customers incurred an increase in the level of setup required for certain features. Call Forwarding subscribers incur the burden of having to activate the service each time they wish to enjoy the potential it offers. 1989 was a major expansion year, marked by the introduction of such features as Call Block, Call Return, Call Selector, and Preferred Call Forwarding. All of these features, while increasing the available potentials, presented additional setup for subscribers.

Figure 6: Burden Categories by Year



IV. Discussion and Future Work

Over the past fifty years, telephony features have evolved dramatically. This is no surprise. The evidence reveals, however, that feature evolution follows well-defined patterns, patterns that are similar to evolution in software architecture [LB85, Leh80] and biological speciation [Fus97, Sne95]. First, the number of features, benefits and burdens all showed a punctuated growth, with major expansions occurring at intervals of a few years. Once the optional features associated with modern telephony were introduced in the early 1970s, expansions occurred every 8-12 years.

Belady and Lehman [LB85] found that the cost and unreliability of software changes rose sharply during a series of minor releases of OS-360, during which changes were locally optimized. Cost and reliability per unit change restabilized when the architecture was redesigned every few releases. Thus the growing entropy of local changes is counteracted by periodic global effort. In biology, too, major speciation events have been explained in terms of the diversity-rewarding effects of major and geologically sudden climatic changes. Such changes, jolts from the outside, encourage rapid diffusion into newly created niches. Punctuated change is also observed in the history of science [Fus97, Sne95], with conceptual reorganizations punctuating epochs of steady, continual growth in knowledge. The feature expansions we observe in telephony can be explained by similar exogenous factors. Whenever major opportunities arose, such as digital switching or legislative changes in US telecommunications regulation, the environment in which the features existed was changed, and a diversification of features followed.

We conclude tentatively therefore that punctuated change is the standard growth process in requirements evolution. Most software systems have not been in existence as long as telephony services, and so it is difficult to generalize about, say, office products, in the same way. Moreover, the serial evolution of systems, so clear in telephony, is overlaid by blending, aggregation and parallel versioning of product lines in many other domains.

It is also clear that features evolve in response to positive and negative feedback influences. Evidence for positive feedback is shown by the steadily non-linear growth in features over time. Over the long term, the rate of feature introduction seems to be a function of the number of features in existence (with the proviso that counting the raw number of features is a very noisy measure of system potential). New capabilities create new ways of working, customer-invented workarounds, and new requirements [Dah93]. Evidence for negative feedback is shown by the dips in features and potential after feature expansions. It appears that now obsolete features hang on for a while before being dropped. Although this may seem intuitively obvious, we have to ask why these features are not kept on as vestiges. Again, biology suggests an answer: vestigial features are expensive to maintain. Once the burden of the feature outweighs its potential advantage relative to alternatives, the feature is no longer viable.

Going beyond sheer numbers, we observe that different types of potential and burden characterize the features of different expansions. What is it that made communication the main emphasis of the first telephony feature expansion but not the remainder? Why privacy next, and then awareness and accessibility? This takes us into the murky area of social history, and we leave it to others to analyze cultural trends, such as the relative change in perceptions of privacy and accessibility. Some general observations seem clear, however. First, it is no accident that the first expansion is dominated by core functional features. Telephony is ultimately all about communicating with others, and so it would be surprising if the first growth in features had been privacy-dominated. We predict that in all applications, the early growth in features is dominated by core functionality: text editing in the case of word processing, account reconciliation in the case of personal finance, and so on. But soon it becomes impossible to differentiate a product and satisfy customers except by supplying features that emphasize other goals: privacy and accessibility, imported graphics, long-term budgeting and financial advice.

Our advice to requirements engineers, is to beware of several intuitive errors. First are overvaluing core features that may not be necessary or fit into the existing use environment and turn out to be gold-plating. The classic example of this in telephony is the communication-enhancing picturephone, which lasted in our corpus of data for precisely one year. These false starts can be analyzed in terms of the profile of potentials versus burdens. Any features that stand out in their potential/burden profile (e.g. by being exceptional in requiring the customer be at a specific place or acquire special equipment that has no other purpose) are likely to be resisted. The second error is to equate more features with greater benefit. Some features do not cohere or impose too great a burden on the customer. We are currently examining the trade-offs between new features and their baseline workarounds to analyze the relative advantage of features in a cohort. The third error is in miscategorizing the system. Telephony started out by being a communication tool. It always will remain that, but it has also become in turn a tool for protecting and invading privacy, a tool for becoming aware of information, and a tool for controlling and defending one's accessibility. Word processors have evolved from being text processing tools to document management tools. Personal finance packages have evolved from being electronic checkbook registers into financial planning assistants. It is impossible to predict with complete accuracy when a major shift in concern will occur during the evolution of a system, but the warning signs are often there. It is possible to ask what the burdens are in a baseline, and this may suggest the emphasis for potentials in a subsequent expansion.

Our classification of potentials and burdens in this study arose using the practices of grounded theory [Gla67]. We think it is important not to invent or propose a universal taxonomy of benefit types for all types of system, but instead to use one's knowledge of a broad domain, including knowledge gained during data collection itself, to shape such a classification scheme. Our goal, after all, is to provide practical help to requirements engineering, not to develop a general theory of value. Nevertheless, a more general scheme would be beneficial in applying our results in other domains. Our classification scheme arose out of earlier work in goal-based requirements engineering [Ant96, Ant99] and the incorporation of analyses of natural-language knowledge and communication verbs [Lev93] mixed with a heavy dose of common sense. We are currently working on a more systematic approach to potentials and burdens that incorporates categories of obstacles and exceptions [Pot99, VL98] into the analysis of features and the workarounds that simulate them.

V. ACKNOWLEDGEMENTS

The authors wish to thank Aamir Hussain of BellSouth Telecommunications, Mary Van Ehrlich of BellSouth Advertising and Publishing Corporation, and the staffs of the Atlanta Library and Atlanta History Center. This research was made possible, in part, due to the co-author's collaboration with the Science & Technology group at BellSouth Telecommunications in Atlanta during the summer of 1999.

VI. REFERENCES

[Ant96] Antón, A.I. "Goal-Based Requirements Analysis," *International Conference on Requirements Engineering (ICRE `96),* Colorado Springs, Colorado, USA, pp. 136-144, April 1996.

[AP98] Antón, A.I. and C. Potts. The Use of Goals to Surface Requirements for Evolving Systems, in *International Conference on Software Engineering (ICSE '98),* Kyoto, Japan, pp. 157-166, 19-25 April 1998.

[Ben97] Bennett, D., *Designing Hard Software: The Essential Tasks.* 1997, Greenwich, Connecticut: Manning Publications / PTR Prentice-Hall.

[Bey98] Beyer, H. and K. Holzblatt, *Contextual Design: Defining Customer-Centered Systems*. 1998, San Francisco, CA: Morgan-Kaufmann.

[BI96] Boehm, B. and H. In. "Identifying Quality Requirement Conflicts," *IEEE Software,* pages 25-35, March 1996.

[CS90] Checkland P.B. and J. Scholes. *Soft Systems Methodology in Action*. Wiley, 1990.

[Dah93] Dahlbom, B. & L. Mathiassen, *Computers in Context : The Philosophy and Practice of Systems Design*, Blackwell, Oxford. 1993

[DeM79] De Marco, T., *Structured Analysis and System Specification*. 1979, Englewood Cliffs, New Jersey: Prentice-Hall.

[Emm89] Emmerson, Andrew. "Strowger and the Invention of the Automatic Telephone Exchange: 1879-1912," 1989.

[Fow97] Fowler, Martin. *UML Distilled: Applying the Standard Object Modeling Notation*, Addison-Wesley, 1997.

[Fus97] Fussy, S; Groessing, G; Schwabl, H, A simple model for the evolution of evolution. *J. Biol. Syst.* ***5***(3): pp. 341-357, 1997

[Gla67] Glaser, B.G. and A.L. Strauss, *The Discovery of Grounded Theory: Strategies for Qualitative Research*. 1967, New York: Aldine De Gruyter.

[Goe97] Goel, A., *et al. MORALE: Mission Oriented Architectural Legacy Evolution*. in *International Conference on Software Maintenance'97.* 1997. Bari, Italy.

[Jac92] Jacobson, I. et. al. *Object-Oriented Software Engineering: A Use Case Driven Approach,* Addison-Wesley, 1992.

[Jac95] Michael Jackson. *Software Requirements and Specifications*. Addison-Wesley, 1995.

[Kar96] Karlson, J. "Software Requirements Prioritizing," In *Proc. 2nd International Conference on Requirements Engineering (ICRE '96)*, pages 110-116, Colorado Springs, Colorado, USA, April 1996.

[KR96] Karlson, J. and K. Ryan. "Prioritizing Software Requirements in an Industrial Setting," In *IEEE International Conference on Systems, Man and Cybernetics*, Vol. 2, pages 953-958, 1996.

[Leh80] Lehman, Manny. "Programs, Life Cycles and the Laws of Software Evolution," *Proc. IEEE Special Issue on Software Evolution*, IEEE, Piscataway, NJ, Vol.68, No. 9, September 1980, pp. 1060-1076.

[LB85] Lehman, M.M. and L.A. Belady. *Program Evolution - Processes of Software Change*, Academic Press, London, 1985.

[Lev93] Levin, Beth. "English Verb Classes and Alternations: A Preliminary Investigation", Chicago, IL: Univ. Chicago Press, 1993

[Mac94] MacDonell, S.G., *Comparative Review of Functional Complexity Assessment Methods for Effort Estimation.* Br. Computer Soc./Inst. Electrical Eng. J., 1994. **9**(3): p. 107-117.

[Nar96] Nardi, B.A., ed. *Context and Consciousness: Activity Theory and Human-Computer Interaction*. . 1996, MIT Press: Cambridge, Massachusetts.

[Nor98] Norman, D.A., *The Invisible Computer : why good products can fail, the personal computer is so complex, and information appliances are the solution.* 1998, Cambridge, Mass.: MIT Press.

[Pot95a] Potts, C. *Invented Requirements and Imagined Customers: Requirements Engineering for Off-the-Shelf Software*. in *RE'95: Second IEEE International Symposium on Requirements Engineering*. 1995. York, UK: IEEE Computer Society Press.

[Pot95b] Colin Potts. Using Schematic Scenarios to Understand User Needs. In *Symposium on Designing Interactive Systems: Processes, Practices, Methods and Techniques*, pages 247-256, University of Michigan, Ann Arbor, Michigan, USA, August 1995.

[Pot99] Potts, C. "A ScenIC: A Strategy for Inquiry-Driven Requirements Determination," Proceedings IEEE 4th International Syposium on Requirements Engineering (RE'99), Limerick, Ireland, 7-11 June 1999.

[Sch98] Schneider, G. and J.P. Winters, *Applying Use Cases: A Practical Guide*. 1998, Reading, Mass.: Addison-Wesley.

[Sne95] Sneppen, K., Bak, P., Flyvbjerg, H., and Jensen, M.H., Evolution as a self-organized critical phenomenon, *Proc. Natl. Acad. Sci. USA,* ***92***(11): pp. 5209-5213, 1995

[Tha92] Thagard, P., *Conceptual Revolutions*. 1992, Princeton, New Jersey: Princeton University Press.

[vLDM95] van Lamsweerde, A. and Darimont, R. and Massonet, P. "Goal-Directed Elaboration of Requirements for a Meeting Scheduler: Problems and Lessons Learnt," Proceedings 2nd International Symposium on Requirements Engineering (RE`95), York, UK, pp. 194-203, March 1995.

[VL98] van Lamsweerde, A. and E. Letier. *Integrating Obstacles in Goal-Driven Requirements Engineering*. in *20th International Conference on Software Engineering*. 1998. Kyoto, Japan: IEEE Computer Society Press.