Location-based Services in Internet Telephony

Xiaotao Wu and Henning Schulzrinne
Department of Computer Science
Columbia University
xiaotaow,hgs@cs.columbia.edu

***Abstract*— Many applications used in the Internet today benefit from using location information. To better handle location information in Internet telephony applications, we did a comprehensive application-layer analysis of location information and location-based communication services. We first summarize and categorize end-user-oriented location description and location detection approaches. We then summarize and categorize how to use location information to provide communication services and introduce several interesting location based communication services. Based on the analysis, we have incorporated location-based service handling in our Session Initiation Protocol (SIP) based Internet telephony infrastructure and our Language for End System Services (LESS).**

I. INTRODUCTION

Location information describes a physical position or attributes of a place, such as place type and privacy status, that may correspond to the past, present or future location of a person, event or device. Many applications used in the Internet today, such as tracking applications, emergency services, ubiquitous computing, and equipment management, benefit from using location information.

In Internet telephony, location information can introduce many new services, not only for tracking, but also for controlling communication behaviors and triggering communication actions. For example, a user agent can automatically adjust its alerting style to vibration in a movie theatre.

In order to better use location information to provide communication services in Internet telephony applications, we need to do a comprehensive analysis of location information and its usages. Previous research work on location-based services [10] give us required technologies to acquire location information and handle the network-layer location-based call routing and QoS management. We believe it is time to analyze location-based services in end-user-oriented manner. Our analysis will not focus on specific location tracking techniques, which have been well defined and analyzed in many articles [10] [26][17][11] [27]. Neither will we discuss the network-layer location-based packet routing and QoS management [3] [4][14][25]. Our focus is on the application-layer, human understandable location descriptions, and end-user-oriented location-based services.

We first investigate how to describe a location and how to get physical locations from end user's point of view in Section II. We then summarize and categorize different location-based services in Section III. Based on the analysis, we extend our Language for End System Services (LESS) [29] to support location-based services in Section IV.

We have built a SIP based Internet telephony infrastructure called Columbia InterNet Extensible Multimedia Architecture (CINEMA) [12] and an intelligent SIP user agent called SIPC [28]. Section V introduces how we integrate location-based services into them.

Section VI concludes the paper and discusses future research works on location-based services.

II. LOCATION DESCRIPTION AND DETECTION

We describe locations in three ways: geospatial coordinates, civil addresses, and location attributes. Geospatial coordinates give the longitude, latitude, and altitude value of a physical location. They are usually used for outdoor location tracking because they can describe a physical position in a unique and standard way. The coordinates are usually acquired by GPS receivers, which do not work indoor. Geospatial coordinates can be encoded in the Geography Markup Language (GML) [5].

Civil addresses provide information similar to postal addresses. For outdoor locations, a civil address can refer to a specific building; for building-level indoor locations, a civil address can refer to a specific room; for room-level indoor locations, a civil address can refer to a specific part of a room, e.g., in the middle of room 123. In the United States, civil addresses can be represented by following the standards published by the United States National Emergency Number Association (NENA) [31]. Civil addresses can also be used for location tracking. Compared to using geospatial coordinates for tracking, civil addresses are easier to understand by end users, but more cumbersome and less accurate for computers to pinpoint a place. Usually, pinpointing a street address on a map requiring human intervention for many causes, e.g., two different places may have the same name, and two different names may represent the same place. There can be a mapping between geospatial coordinates and civil addresses. The mapping can help people to choose proper formats for location-based services when only one type of location information available. Another common usage of civil addresses is for resource discovery, e.g., to find nearby restaurants or available devices, like printers.

Location attributes are used to describe factors of a location that may affect communication behaviors, e.g., the place type, or the number of people inside a certain area. Location attributes can be applied to both indoor and outdoor locations. There are no standards defining location attributes, but the IETF Rich Presence Format [23] draft has provided some commonly used attributes, such as the type and privacy status of a place. Usually, we use location attributes to make communication decisions, e.g., rejecting incoming calls requiring privacy when the caller is in a public place.

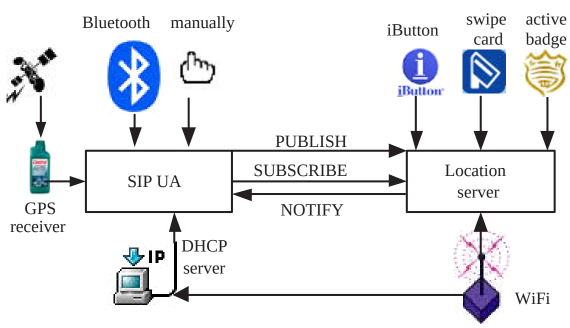

Fig. 1. Location detection

Different location descriptions require different location detection technologies. As shown in Figure 1, geospatial coordinates are usually acquired by using GPS receivers. They can be transmitted through serial port or other I/O interfaces to users' communication agents. People can store civil address information or location attributes of a room in a Bluetooth device. When a communication agent with Bluetooth support enters the room, it can get the location information of the room through BlueTooth beacons. Both civil address and geospatial address information of a user can be stored in local DHCP server and transmitted by DHCP options [18][20].

The above approaches have the location information directly sent to users' communication agents, and the agents associate users' identities to their locations. We name this approach agent-centric location detection. In a SIP based Internet telephony system, the result of the agent-centric location detection can be sent to location servers in SIP PUBLISH requests [16].

Location detection can also be server-centric when communication agents cannot get their locations directly. In Figure 1, a user can put his profile in a small device, such as a swipe card, an IR/RF programmable badge or an i-Button [24]. The device readers in a context can read the user's profile and send the profile to a location server. The location server knows the device reader's location and will associate the profile with the location. In SIP based Internet telephony systems, SIP user agents may subscribe to their own location events by following the SIP event notification architecture [19].

Table I shows the differences between two approaches for location detection.

	Server-centric	Agent-centric
User devices	Cheaper	More expensive
Privacy	Limited	Better control
Setup	Pre-knowledge of users' profiles	No pre-knowledge required.

TABLE I
DIFFERENCES BETWEEN TWO MODES

Usually, user devices used in the server-centric approach are cheaper than those in the agent-centric approach. A swipe card or a RF badge may cost less than one dollar, but a BlueTooth device or a GPS receiver is more expensive. Using the agent-centric approach, users may have better control over their location privacy, while using server-centric approach, the control is limited to what the server can offer. The server-centric approach usually requires pre-knowledge of users' profiles to map device IDs to users' URIs.

In a large social event, such as a big conference, since people come to communicate with each other, they are more likely to release their location information and the location privacy is not an essential concern. In addition, people usually need to register to join a conference so a location server can easily get the participants' profiles. Hence, the server-centric mode is an economic way to handle location detection for big conferences.

For a place often having visitors, such as a hotel room, the agent-centric approach is more appropriate. Users do not have to provide their profiles beforehand for location detection. The location information will be stored only in users' own communication agents so that users can fully control their location privacy.

III. LOCATION-BASED SERVICES

Location-based services can be divided into five categories. First, we may send location information to remote parties. This set of services are commonly used today, e.g., in location tracking applications. Second, use location information to make communication decisions, e.g., a user agent may automatically disable instant messaging when driving. Third, location changes can trigger communication actions, e.g., when a person's user agent gets a location notification indicating the person enters a room, the user agent may automatically turn on the light of the room. Fourth, we may use location information in resource discovery, e.g., we can put location information in a Service Location Protocol (SLP) [8] query to find available multimedia input/output devices nearby. Fifth, treat a location as a communication entity, e.g., an instant message to `sip:room123@examples.com` will be broadcasted to all the people in `room 123`. We will discuss each set of services in detail below.

A. Sending location information to remote parties for location tracking

Locations are usually represented in geospatial coordinates or civil addresses for tracking. In SIP based Internet telephony systems, location tracking is based on the SIP event notification architecture [19]. A watcher sends a SIP `SUBSCRIBE` request to a presentity. Once the subscription gets accepted, the presentity can then use SIP `NOTIFY` requests to send the location information to the watcher. The safety and privacy issues are important for location tracking. These issues have been addressed in IETF Geopriv working group drafts [6][22] [21].

Another way to send location information is to put the information in a SIP `INVITE`, `INFO` or `UPDATE` request, encoded in MIME multipart format [7]. This can be used in emergency call handling. When an Emergence Call Center (ECC) receives an emergency call, it can pinpoint where the caller is based on the location information in the `INVITE` request.

B. Making communication decisions

Different locations may require different communication behaviors. For example, video or text conversation is not good when driving. User agents usually based on location attributes, instead of geospatial coordinates or civil addresses, to choose appropriate communication behaviors. Either caller's location or callee's location may affect communication behaviors. Other information, such as calendar information, may be combined with location information to deduce appropriate communication behaviors. User agents should respect the required communication behaviors when making communication decisions. We illustrate different kinds of location-based decisions below based on who makes the decision and the source of the location information.

Caller makes decision based on caller's location:
: When making an outgoing call, a caller's user agent may check its own location to decide how to handle the outgoing call. For example, if the caller is in a place requiring quiet, the caller's user agent may only enable text and video conversation, and mute audio devices.

Caller makes decision based on callee's location:
: A caller may watch a callee's location and make call decisions based on that. For example, if the callee is driving, the caller's user agent may suggest the caller to call sometime later.

Callee makes decision based on caller's location:
: A callee may also watch a caller's location for call decision making. For example, if the callee prefers to have private conversation, callee's user agent may check the caller's location privacy status. Location privacy status indicates whether third parties may be able hear or view any parts of a conversation. The value of privacy status can be `public` or `private` [23]. If the callee finds the caller's location privacy status is `public`, it may reject the call.

Callee makes decision based on callee's location:
: The callee's user agent can check its own location for incoming call handling. For example, if the callee is in a place requiring quiet, the callee's user agent may choose to vibrate the device for incoming calls.

Call decisions based on both caller and callee's locations:
: In many cases, both the caller and the callee's locations are taken into account for communication decision making. For example, a private conversation requires both the caller and the callee's location privacy status to be `private`.

Combine location information and other information:
: Location information can be combined with other information, such as calendar information, to deduce appropriate communication behaviors. For example, in a conference, when a session is going on, the room of the session should be quiet. By checking the conference calendar, a user agent may know whether its current location requiring quiet or not.

C. Triggering actions

User agents may invoke actions when detecting location changes. Location changes can be in an incoming location notification from a location server, or retrieved through locally connected location sensors. We divide this set of services into three classes based on the source of the location changes.

Actions triggered by user's own location changes:
: For example, when a user drives on the way to his office, his user agent may get a location notification and automatically turn on the air-conditioner in his office. Another example, when a user moves from one location to another, his user agent may transfer the ongoing media session to the user's new location [1]. For this set of services, users subscribe to their own location information. There is no authorization needed.

Actions triggered by remote parties' location changes:
: For example, in a day care center, when a child leaves the playground, the teacher may

get called. For this set of services, users subscribe to others' location information and need to get authorization for acquiring the location information.

Actions triggered by location relationship changes: For example, when two friends are close to each other on the street, their user agents may automatically make a call or send an instant message to each other so they may not miss each other.

D. Resource discovery

Location information can be used in resource discovery queries to find nearby resources. This kind of services is commonly used in map services on the web. A user inputs a civil address, map services may return a list of nearby restaurants and points of interest. In Internet telephony systems, a user agent with limited multimedia I/O capabilities may put location information in a Service Location Protocol (SLP) query to find available multimedia input/output resources in the context. The user agent can then control the resources for multimedia call handling [2].

E. Treat a location as a communication entity

We can assign a URI to a location and treat the location as a communication entity. We may use the URI to represent all the people in that location. For example, A person may send an instant message or an email to the location URI. The instant message or the email will get broadcasted to all the people in that location. A person may also invite all the people in that location to a conference by simply sending an invitation to the location URI. Users can also subscribe to a location URI to acquire location attributes, such as the number of people in the location. The location attributes may help to make communication decisions or trigger communication actions. For example, start a conversation with Alice, who is in `room 123`, only if the number of people in `room 123` is 1, turn off the light of `room 123` if the number of people in the room is changed to zero.

IV. Extending the Language for End System Services for location-based services

We defined the Language for End System Services (LESS) [29] to handle services in intelligent end systems, such as SIPC [28]. LESS is extended from the Call Processing Language (CPL) [13], which is designed to handle services residing on network servers, such as SIP proxy servers. LESS is an XML-based language, uses a tree-like structure to make communication decisions. The tree-like structure makes it easy to convert a LESS script to a graphical representation and vice versa. LESS is not designed to handle all kinds of services, but to handle most commonly used communication features with keeping the language simple, safe, easy to understand by non-programmers, and easy to analyze for feature interactions. LESS is not a Turing-complete language. There is no loop, and no user-defined variables in LESS.

In LESS, `switch`es are used to represent communication decisions a script can make. For example, Figure 2 shows a LESS script handling the call screening service by using the `address-switch`. In the script, the `address-switch` checks the caller's address, if it matches `sip:bob@example.com`, the LESS script will instruct the user agent to reject the call, otherwise, it will automatically `accept` the call.

```
<incoming>
  <address-switch field="origin">
    <address is="sip:bob@example.com">
      <reject status="486"/>
    </address>
    <otherwise>
      <accept/>
    ....
```

Fig. 2. Call screening in LESS

In its original design, LESS cannot handle location-based services. We extend the language for location information handling, but still keep the tree-like structure of the language. The extension is based on the above analysis of location information and its usages.

```
<address-switch field="origin">
 <address is="sip:bob@example.com">
  <notify>
   <cp:transformations>
    <gp:civil-loc-transformation>
     full
    </gp:civil-loc-transformation>
   </cp:transformations>
  </notify>
 </address>
   ....
```

Fig. 3. Location notification

To send location information to remote parties, we extend the LESS `notify` command to support location information. Different user groups may need different location privacy policies. By following the IETF Geopriv Policy draft [21], we allow the LESS `notify` command to have new elements of geopriv policies. Figure 3 shows an example. In the example, only `sip:bob@example.com` can get the full civil location notification.

We defined `location-switch` [30] for location-based communication decision making. For a LESS script, location information can be the script owner's location, the remote party's location, or relationship between multiple locations. If combined with other switches, such as time-switch and address-switch, we can integrate location information and other information to provide services.

```
<incoming>
  <time-switch>
    <time dtstart="20040618T150000Z"
          dtend="20040618T160000Z">
      <location-switch type="civil">
        <location LOC="conf-room">
          <reject status="busy"/>
        </location>
      ....
```

Fig. 4. Location-based call rejection

Figure 4 shows how to use `location-switch` to make communication decisions. In this example, in the time period between `3:00:00PM` and `4:00:00PM` on Jun 18, if the user is in `conf-room`, all incoming calls will get rejected automatically.

V. Implementation

We have built a SIP based Internet telephony infrastructure called Columbia InterNet Extensible Multimedia Architecture (CINEMA) [12] and an intelligent SIP user agent called SIPC [28]. To integrate location-based services into our SIP-based Internet telephony systems, instead of implementing the services one-by-one, we chose to build a CPL engine in our CINEMA infrastructure and a LESS engine in SIPC and provide the support for `location-switch` and location notifications in the CPL and the LESS engines. By this way, users can easily create new services without changing our implementations. Many location-based services mentioned in Section III can be handled by LESS. We briefly introduce the LESS engine and the LESS service creation environment below.

When an event happens in SIPC, e.g., receiving an incoming call, SIPC's LESS engine will first initialize its Service Logic Execution Environment (SLEE) by collecting information of the call and the context. It will then check the LESS script loaded into SIPC. If the script can handle the event, the LESS engine will perform the decision tree traversal based on the script. For each `switch` met in the tree traversal, the engine matches the `switch` against the collected information to make decisions. For example, for location-based services, the LESS engine collects location information from the location sensors and the location notifications. If there is a `location-switch` in the service script, the engine will check whether the location defined in the `location-switch` matches the collected location information and make decisions.

We have built a graphical service creation environment with location-based service support. Figure 5 shows a graphical representation of the LESS script in Figure 4. The `location-switch` in the figure helps to make call decisions based on the script owner's location.

Since LESS is designed to be simple and easy to understand by end users, its functionalities are limited. Some location-based services have to be programmed in a general Turing-complete programming language, such as C/C++ or Java. Location tracking in emergency call handling is such a service.

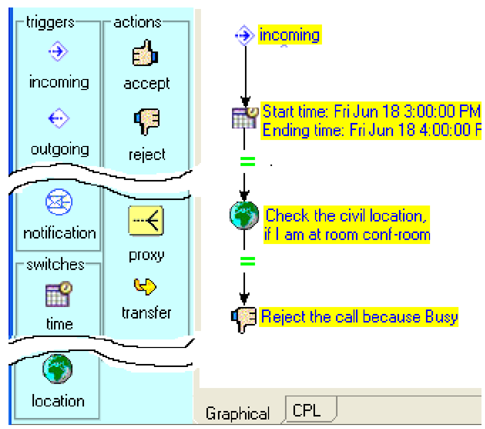

Fig. 5. Location-based service creation

We are developing an emergency call handling architecture which requires location-based services. Figure 6 shows the architecture.

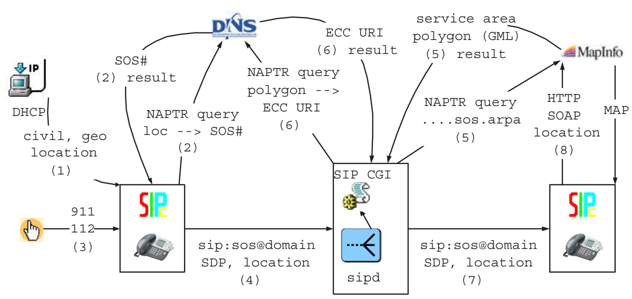

Fig. 6. SOS call handling architecture

We can roughly divide the emergency call handling system into three parts. The SIP user agents in the left part of the architecture handling step (1) (2) (3) are used to identify emergency calls. The emergency numbers are different in different countries. For example, in U.S., the emergency number is 911, but in Sweden, it is 112. The first three steps help to identify emergency numbers based on user's location information.

In the architecture, SIPD, the proxy server in our CINEMA infrastructure, uses SIP CGI scripts to handle step (4) (5) (6) to route calls. When SIPD receives an emergency call, it first gets the location information of the caller. It then tries to query a database to get the service area based on the location information. When the server gets the service area information, usually a polygon described in Geography Markup Language (GML) [5], the server will send a DNS query for the NAPTR [15] records of the Emergency Call Center (ECC) URI of the service area. The proxy server will then forward the call to the ECC URI.

The SIP user agents in the right part of the architecture are used to take emergency calls and pinpoint caller's position in a map.

VI. Conclusion

In this paper, we summarize and categorize different location-based communication services. We then extend our Language for End System Services (LESS) to support location-based services. Location information gets used more and more often in people's daily life. This paper only focuses on communication related location-based services. In Internet telephony systems, communication services can be enhanced by the integration of other Internet services, such as email, web, and network gaming, which also involve location information handling. Further investigation should be conducted on how to provide more innovative location-based services with the integration of other Internet services.

References

[1] Jean Bacon, John Bates, and David Halls. Location-oriented multimedia. *IEEE Personal Communications Magazine*, 4(5), October 1997.

[2] Stefan Berger, Henning Schulzrinne, Stylianos Sidiroglou, and Xiaotao Wu. Ubiquitous computing using SIP. In *ACM NOSSDAV 2003*, June 2003.

[3] Tracy Camp, Jeff Boleng, and Lucas Wilcox. Location information services in mobile ad hoc networks. In *Conference Record of the International Conference on Communications (ICC)*, pages 3318–3324, New York, New York, April 2002.

[4] Tracy Camp, Jeff Boleng, Brad Williams, William Navidi, and Lucas Wilcox. Performance comparison of two location based routing protocols for ad hoc networks. In *Proceedings of the Conference on Computer Communications (IEEE Infocom)*, New York, New York, June 2002.

[5] Open GIS Consortium. Geography markup language. http://xml.coverpages.org/GMLv310-20040207-d20040326.pdf.

[6] M. Danley, D. Mulligan, J. Morris, and J. Peterson. Threat analysis of the geopriv protocol. RFC 3694, Internet Engineering Task Force, February 2004.

[7] N. Freed and N. Borenstein. Multipurpose Internet mail extensions (MIME) part one: Format of Internet message bodies. RFC 2045, Internet Engineering Task Force, November 1996.

[8] E. Guttman, C. E. Perkins, J. Veizades, and M. Day. Service location protocol, version 2. RFC 2608, Internet Engineering Task Force, June 1999.

[9] Jeffrey Hightower and Gaetano Borriella. Location systems for ubiquitous computing. *IEEE Computer*, 34(8):57–66, 2001.

[10] Jeffrey Hightower and Gaetano Borriello. A survey and taxonomy of location sensing systems for ubiquitous computing. Technical Report UW CSE 01-08-03, University of Washington, Department of Computer Science and Engineering, Seattle, Washington, August 2001.

[11] Jeffrey Hightower, Barry Brumitt, and Gaetano Borriello. The location stack: A layered model for location in ubiquitous computing. In *IEEE Workshop on Mobile Computing Systems & Applications (WMCSA 2002)*, pages 22–28, Callicoon, NY, June 2002. IEEE.

[12] Wenyu Jiang, Jonathan Lennox, Sankaran Narayanan, Henning Schulzrinne, Kundan Singh, and Xiaotao Wu. Integrating Internet telephony services. *IEEE Internet Computing*, 6(3):64–72, May 2002.

[13] Jonathan Lennox, Xiaotao Wu, and Henning Schulzrinne. CPL: a language for user control of Internet telephony services. Internet draft, Internet Engineering Task Force, August 2003. Work in progress.

[14] Yunhao Liu, Xiaomei Liu, Li Xiao, Lionel Ni, and Xiaodong Zhang. Location-aware topology matching in P2P systems. In *IEEE Infocom 2004*, March 2004.

[15] M. Mealling and R. W. Daniel. The naming authority pointer (NAPTR) DNS resource record. RFC 2915, Internet Engineering Task Force, September 2000.

[16] A. Niemi. Session initiation protocol (SIP) extension for event state publication. Internet Draft draft-ietf-sip-publish-02, Internet Engineering Task Force, January 2004. Work in progress.

[17] Kaveh Pahlavan, Xinrong Li, Mika Ylianttila, Ranvir Chana, and Matti Latva-aho. An overview of wireless indoor geolocation techniques and systems. *Lecture Notes in Computer Science*, 1818:1–13, May 2000.

[18] J. Polk, J. Schnizlein, and M. Linsner. Dynamic host configuration protocol option for coordinate-based location configuration information. Internet Draft draft-ietf-geopriv-dhcp-lci-option-03, Internet Engineering Task Force, December 2003. Work in progress.

[19] A. B. Roach. Session initiation protocol (sip)-specific event notification. RFC 3265, Internet Engineering Task Force, June 2002.

[20] Henning Schulzrinne. DHCP option for civil location. Internet draft, Internet Engineering Task Force, July 2003. Work in progress.

[21] Henning Schulzrinne. Policy rules for disclosure and modification of geographic information. Internet Draft draft-ietf-geopriv-policy-00, Internet Engineering Task Force, November 2003. Work in progress.

[22] Henning Schulzrinne. Common policy. Internet Draft draft-ietf-geopriv-common-policy-00, Internet Engineering Task Force, February 2004. Work in progress.

[23] Henning Schulzrinne. RPID – rich presence information data format. Internet Draft draft-ietf-simple-rpid-01, Internet Engineering Task Force, February 2004. Work in progress.

[24] Dallas Semiconductor. iButton. http://www.ibutton.com.

[25] S. H. Shah and Klara Nahrstedt. Predictive location-based QoS routing in mobile ad hoc networks. In *Conference Record of the International Conference on Communications (ICC)*, pages 1022–1027, New York, NY, USA, April 2002.

[26] M. van Steen, F. J. Hauck, Philip Homburg, and A. S. Tanenbaum. Locating objects in wide-area systems. *IEEE Communications Magazine*, 36(1):104–109, January 1998.

[27] Roy Want, Andrew Hopper, Veronica Falcao, and J. A. Gibbons. The active badge location system. *ACM Transactions on Information Systems*, 10(1):91–102, January 1992. also Olivetti Research Limited Technical Report ORL 92-1.

[28] Xiaotao Wu. Columbia university sip user agent (sipc). http://www.cs.columbia.edu/IRT/sipc.

[29] Xiaotao Wu and Henning Schulzrinne. Programmable end system services using SIP. In *Conference Record of the International Conference on Communications (ICC)*, May 2003.

[30] Xiaotao Wu and Henning Schulzrinne. Location-switch for call processing language (CPL). Internet draft, Internet Engineering Task Force, February 2004. Work in progress.

[31] National Emergency Number Association (NENA) www.nena.org. Nena technical information document on model legislation enhanced 911 for multi-line telephone systems. nov 2000. http://www.nena.org/9-1-1TechStandards/TechInfoDocs/MLTS_ModLeg_Nov2000.PDF.